Figure 1.3 Reece's Slip-Sinkage Theory

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file - Votre référence

Votre file - Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

Analysis of the Tractive Performance of Pneumatic Tires over Soft Terrain

by

SONG-TAO CHEN, B Eng., M.Eng.

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfilment of
the requirements for the degree of
Doctor of Philosophy

Ottawa-Carleton Institute for Mechanical and Aerospace Engineering
Department of Mechanical and Aerospace Engineering
Carleton University
Ottawa, Ontario
August 1993

© Copyright
1993, Song-tao Chen

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file - Votre référence

Our file - Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-89735-X

Canada

Name **SONG-TAO CHEN**

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

MECHANICAL ENGINEERING

SUBJECT TERM

0548

SUBJECT CODE

U·M·I

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS

Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION

General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Finance 0277
Guidance and Counseling 0519
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523

Psychology 0525
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS

Language
General 0679
Ancient 0289
Linguistics 0290
Modern 0291
Literature
General 0401
Classical 0294
Comparative 0295
Medieval 0297
Modern 0298
African 0316
American 0591
Asian 0305
Canadian (English) 0352
Canadian (French) 0355
English 0593
Germanic 0311
Latin American 0312
Middle Eastern 0315
Romance 0313
Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY

Philosophy 0422
Religion
General 0318
Biblical Studies 0321
Clergy 0319
History of 0320
Philosophy of 0322
Theology 0469

SOCIAL SCIENCES

American Studies 0323
Anthropology
Archaeology 0324
Cultural 0326
Physical 0327
Business Administration
General 0310
Accounting 0272
Banking 0770
Management 0454
Marketing 0338
Canadian Studies 0385
Economics
General 0501
Agricultural 0503
Commerce-Business 0505
Finance 0508
History 0509
Labor 0510
Theory 0511
Folklore 0358
Geography 0366
Gerontology 0351
History
General 0578

Ancient 0579
Medieval 0581
Modern 0582
Black 0328
African 0331
Asia, Australia and Oceania 0332
Canadian 0334
European 0335
Latin American 0336
Middle Eastern 0333
United States 0337
History of Science 0585
Law 0398
Political Science
General 0615
International Law and Relations 0616
Public Administration 0617
Recreation 0814
Social Work 0452
Sociology
General 0626
Criminology and Penology 0627
Demography 0628
Ethnic and Racial Studies 0631
Individual and Family Studies 0628
Industrial and Labor Relations 0629
Public and Social Welfare 0630
Social Structure and Development 0700
Theory and Methods 0344
Transportation 0709
Urban and Regional Planning 0999
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES

Agriculture
General 0473
Agronomy 0285
Animal Culture and Nutrition 0475
Animal Pathology 0476
Food Science and Technology 0359
Forestry and Wildlife 0478
Plant Culture 0479
Plant Pathology 0480
Plant Physiology 0817
Range Management 0777
Wood Technology 0746
Biology
General 0306
Anatomy 0287
Biostatistics 0308
Botany 0309
Cell 0379
Ecology 0329
Entomology 0353
Genetics 0369
Limnology 0793
Microbiology 0410
Molecular 0307
Neuroscience 0317
Oceanography 0416
Physiology 0433
Radiation 0821
Veterinary Science 0778
Zoology 0472
Biophysics
General 0786
Medical 0760

EARTH SCIENCES

Biogeochemistry 0425
Geochemistry 0996

Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES

Environmental Sciences 0768
Health Sciences
General 0566
Audiology 0300
Chemotherapy 0992
Dentistry 0567
Education 0350
Hospital Management 0769
Human Development 0758
Immunology 0982
Medicine and Surgery 0564
Mental Health 0347
Nursing 0569
Nutrition 0570
Obstetrics and Gynecology 0380
Occupational Health and Therapy 0354
Ophthalmology 0381
Pathology 0571
Pharmacology 0419
Pharmacy 0572
Physical Therapy 0382
Public Health 0573
Radiology 0574
Recreation 0575

Speech Pathology 0460
Toxicology 0383
Home Economics 0386

PHYSICAL SCIENCES

Pure Sciences
Chemistry
General 0485
Agricultural 0749
Analytical 0486
Biochemistry 0487
Inorganic 0488
Nuclear 0738
Organic 0490
Pharmaceutical 0491
Physical 0494
Polymer 0495
Radiation 0754
Mathematics 0405
Physics
General 0605
Acoustics 0986
Astronomy and Astrophysics 0606
Atmospheric Science 0608
Atomic 0748
Electronics and Electricity 0607
Elementary Particles and High Energy 0798
Fluid and Plasma 0759
Molecular 0609
Nuclear 0610
Optics 0752
Radiation 0756
Solid State 0611
Statistics 0463
Applied Sciences
Applied Mechanics 0346
Computer Science 0984

Engineering
General 0537
Aerospace 0538
Agricultural 0539
Automotive 0540
Biomedical 0541
Chemical 0542
Civil 0543
Electronics and Electrical 0544
Heat and Thermodynamics 0348
Hydraulic 0545
Industrial 0546
Marine 0547
Materials Science 0794
Mechanical 0548
Metallurgy 0743
Mining 0551
Nuclear 0552
Packaging 0549
Petroleum 0765
Sanitary and Municipal 0554
System Science 0790
Geotechnology 0428
Operations Research 0796
Plastics Technology 0795
Textile Technology 0994

PSYCHOLOGY

General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

The undersigned hereby recommend to
the Faculty of Graduate Studies and Research
acceptance of the thesis,

**ANALYSIS OF THE TRACTIVE PERFORMANCE
OF PNEUMATIC TIRES OVER SOFT TERRAIN**

submitted by
SONG-TAO CHIEN, B.ENG., M.ENG.

in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

Chairman,
Department of Mechanical and Aerospace Engineering

Thesis Supervisor

External Examiner

Carleton University
September 1993

Abstract

The prediction and evaluation of the tractive performance of off-road wheeled vehicles is one of the major topics in Terramechanics. To predict their performance, it is essential to have a basic understanding of the mechanics of tire-terrain interaction. Two tire-terrain interaction models are presented in this thesis. One is a practical model with relatively simple formulation, and the other is a three-dimensional finite element model. Both the practical and the finite element tire-terrain interaction models make use of the data obtained from static tire tests, or from the "Generalized Deflection Chart" provided by tire manufacturers to characterize tire behaviour. For both models, the terrain is represented by a group of semi-empirical equations describing the pressure-sinkage relationship, the response of terrain to repetitive loadings, the shear stress-shear displacement relationship, and the slip sinkage-shear displacement relationship. These relationships are obtained using the bevameter technique.

The slip-sinkage effect is considered, and approaches for incorporating slip-sinkage into tire-terrain interaction models are investigated. Basic equilibrium conditions are not sufficient to completely define the characteristics of tire-terrain interaction if slip-sinkage is taken into account, and an additional condition or relationship is required. The method of two-equilibrium-position assumption and that of work equivalence assumption are studied.

Experimental validation of the practical and the finite element models is carried out using the forced-slip tire tester available at the Transport Technology Research Laboratory, Carleton University. Both the measured and the predicted tractive performances using both models, including thrust, motion resistance, drawbar pull and sinkage, are presented. It was found that for tires with high inflation pressures and operating over relatively soft terrain, the practical model is reasonably good for predicting tractive performance, particularly at slips equal to or greater than 20 %. For tires with low inflation pressures and operating over relatively firm terrain, the finite element model gives better predictions, primarily due to a more realistic modelling of tire-terrain contact geometry. The results of this investigation also indicate that the slip-sinkage effect is an important factor that should be taken into account when modelling the performance of a tire operating on frictional terrain, such as dry sand.

Acknowledgements

The author wishes to extend his deepest gratitude and appreciation to his thesis supervisor, Professor J.Y. Wong, for his invaluable guidance, encouragement, and suggestions throughout the course of this investigation and in preparing the thesis.

The financial support provided by the Natural Sciences and Engineering Research Council of Canada through grants awarded to Professor J.Y. Wong is gratefully acknowledged. The scholarship awarded by Carleton University is also greatly appreciated.

The author also wishes to sincerely thank Dr. H. Nakashima, formerly Post-doctoral Fellow, Transport Technology Research Laboratory, Carleton University, and now with Department of Agricultural Engineering, Kyoto University, Japan, for providing his computer codes for simulating tire-hard ground interaction.

Finally, the author takes this opportunity to express his gratitude and appreciation to his wife, Haiyan, for her love, patience and support, and to his family members for their continuing support.

Table of Contents

Acceptance Sheet	ii
Abstract	iii
Acknowledgements	iv
Table of Contents	v
List of Tables	viii
List of Figures	x
Principal Symbols	xiii
1 Literature Review	1
1.1 Review of Tire-Terrain Interaction Models	1
1.2 Review of Finite Element Tire Models on Hard Surface	9
1.3 Review of Studies of Slip-Sinkage	11
1.4 Scope of the Present Work	13
2 Finite Element Model of Pneumatic Tires	16
2.1 Finite Element Representation of Tires	16
2.2 Derivation of the Young's Moduli	18
2.3 Static Testing of Pneumatic Tires	23
2.4 Verification of the Finite Element Tire Model	29
3 Semi-Empirical Model of Terrain	35
3.1 Bevameter Tests	35
3.1.1 Pressure-Sinkage Test	36
3.1.2 Torque-Shear Displacement Test	40
3.2 Methods for Processing Bevameter Data	40
3.2.1 Pressure-Sinkage Relationship	40
3.2.2 Shear Stress-Shear Displacement Relationship	46

3.2	Methods for Processing Bevameter Data	40
3.2.1	Pressure-Sinkage Relationship	40
3.2.2	Shear Stress-Shear Displacement Relationship	46
3.2.3	Slip Sinkage-Shear Displacement Relationship	48
4	Tire-Terrain Interaction Models	53
4.1	Practical Model	53
4.1.1	Prediction of Sinkage and Shear Displacement	55
4.1.2	Prediction of Slip-Sinkage for Tire	58
4.1.3	Criterion for Determining the Operating Mode of a Tire	80
4.1.4	Prediction of Shear Stress under Continuously Varying Normal Pressure	81
4.1.5	Work Equivalence Assumption and Search Scheme	86
4.2	Finite Element Model	89
4.2.1	Conversion of the Normal Pressure to Nodal Forces	89
4.2.2	Conversion of the Shear Stress to Nodal Forces	92
4.2.3	Calculation of Shear Displacement	94
4.2.4	Slip-Sinkage Model	100
4.2.5	Incremental-Iterative Scheme and Convergence Criterion	101
4.2.6	Verification of the Incremental-Iterative Scheme	104
5	Experimental Verification of Tire-Terrain Interaction Models	107
5.1	Forced-Slip Tire Tester	107
5.2	Experiments	113
5.3	Numerical Examples	126
6	Conclusions and Recommendations	161
6.1	Conclusions	161
6.2	Recommendations	163

References	166
Appendices	172
A Formulation of Least Squares Method for Fitting Nonlinear Expressions	172
B Computer Source Codes for Simulation of Tire-Hard Ground Interaction — Finite Element Model	174
C Computer Source Codes for Simulation of Tire-Soft Terrain Interaction — Practical Model	216
D Computer Source Codes for Simulation of Tire-Soft Terrain Interaction — Finite Element Model	233
E Computer Graphics Source Codes for Finite Element Models	265
F Experimental Observations on Normal Pressure and Shear Stress Distributions	296

List of Tables

2.1	Michelin 11.00R16 XL Tire Dimensions	25
3.1	Pressure-Sinkage Parameters	16
3.2	Shear Parameters	17
3.3	Slip-Sinkage Parameters for Eq. (3.16)	18
3.4	Slip-Sinkage Parameters for Eq. (3.18)	50
4.1	Results of Test Case for Verifying the Incremental Iterative Scheme .	106
5.1	Measured Thrusts at 30 psi (206.83 kPa)	113
5.2	Measured Resistances at 30 psi (206.83 kPa)	120
5.3	Measured Drawbar Pulls at 30 psi (206.83 kPa)	120
5.4	Measured Total Sinkages at 30 psi (206.83 kPa)	121
5.5	Measured Thrusts at 20 psi (137.89 kPa)	121
5.6	Measured Resistances at 20 psi (137.89 kPa)	122
5.7	Measured Drawbar Pulls at 20 psi (137.89 kPa)	122
5.8	Measured Total Sinkages at 20 psi (137.89 kPa)	123
5.9	Measured Thrusts at 10 psi (68.94 kPa)	123
5.10	Measured Resistances at 10 psi (68.94 kPa)	124
5.11	Measured Drawbar Pulls at 10 psi (68.94 kPa)	124
5.12	Measured Total Sinkages at 10 psi (68.94 kPa)	125
5.13	Finite Element Simulation Results at 30 psi (206.83 kPa)	126
5.14	Finite Element Simulation Results at 20 psi (137.89 kPa)	126
5.15	Finite Element Simulation Results at 10 psi (68.94 kPa)	128
5.16	Predicted Performance Using the Practical Model	128

5.17 Comparison of Predicted and Measured Thrusts	
at 30 psi (206.83 kPa)	141
5.18 Comparison of Predicted and Measured Resistances	
at 30 psi (206.83 kPa)	141
5.19 Comparison of Predicted and Measured Drawbar Pulls	
at 30 psi (206.83 kPa)	142
5.20 Comparison of Predicted and Measured Total Sinkages at	
30 psi (206.83 kPa)	142
5.21 Comparison of Predicted and Measured Thrusts at	
20 psi (137.89 kPa)	143
5.22 Comparison of Predicted and Measured Resistances at	
20 psi (137.89 kPa)	143
5.23 Comparison of Predicted and Measured Drawbar Pulls at	
20 psi (137.89 kPa)	144
5.24 Comparison of Predicted and Measured Total Sinkages at	
20 psi (137.89 kPa)	144
5.25 Comparison of Predicted and Measured Thrusts at	
10 psi (68.94 kPa)	145
5.26 Comparison of Predicted and Measured Resistances at	
10 psi (68.94 kPa)	145
5.27 Comparison of Predicted and Measured Drawbar Pulls at	
10 psi (68.94 kPa)	146
5.28 Comparison of Predicted and Measured Total Sinkages at	
10 psi (68.94 kPa)	146
5.29 Regression Coefficients for All Fitted Performance Curves	147
5.30 Comparison of Predicted and Measured Tractive Performance	
at 20 % Slip	149

List of Figures

1.1	Operating Modes in Wong's Tire Model	4
1.2	Bekker's Slip-Sinkage Model	12
1.3	Reece's Slip-Sinkage Theory	14
2.1	Finite Element Tire Model for Simulating Static Test	17
2.2	Method for Searching for Young's Modulus Ratio α	24
2.3	Schematic of Static Tire Tester	24
2.4	Block Diagram of Data Acquisition System for Static Tire Test	26
2.5	Measured Load-Deflection Data	27
2.6	Measured Contact Area-Deflection Data	28
2.7	Derived Young's Moduli for 30 psi (206.83 kPa) Inflation Pressure	30
2.8	Derived Young's Moduli for 20 psi (137.89 kPa) Inflation Pressure	31
2.9	Derived Young's Moduli for 10 psi (68.94 kPa) Inflation Pressure	32
2.10	Measured and Predicted Load-Deflection Curves	33
2.11	Measured and Predicted Contact Area-Deflection Curves	34
3.1	Schematic of Bevameter	37
3.2	Block Diagram of Data Acquisition System for Bevameter Test	38
3.3	Sample of Measured Pressure-Sinkage Data	39
3.4	Sample of Measured Internal Shear Stress-Shear Displacement Data	41
3.5	Sample of Measured Internal Slip-Sinkage-Shear Displacement Data	42
3.6	Sample of Measured Rubber Shear Stress-Shear Displacement Data	43
3.7	Sample of Measured Rubber Slip-Sinkage-Shear Displacement Data	44
3.8	Internal Slip-Sinkage-Shear Displacement Data Fitted with Eq. (3.18)	51
3.9	Rubber Slip-Sinkage-Shear Displacement Data Fitted with Eq. (3.18)	52
4.1	Illustration of Operating Modes for Practical Tire-Terrain Interaction Model	54
4.2	Illustration of Terrain Repetitive Loading Characteristics	56

4.3	Calculation of Shear Displacement for Practical Tire Model	57
4.4	Illustration of Different States for Two-Equilibrium-Position Approach	60
4.5	Method for Calculating Slip-Sinkage Using Normal Component	63
4.6	Illustration of Slip-Sinkage for a Rigid Plate	66
4.7	Methods for Calculating Slip-Sinkage for Rigid Mode	68
4.8	Illustration of Limits on Methods for Calculating Slip-Sinkages	69
4.9	Definition of Slip-Sinkage for Rigid Mode	72
4.10	Comparisons of Slip-Sinkages for $Z = 5$ cm and $Z_0 = 1$ cm	75
4.11	Comparisons of Slip-Sinkages for $Z = 5$ cm and $Z_0 = 5$ cm	76
4.12	Comparisons of Slip-Sinkages for $Z = 5$ cm and $Z_0 = 12$ cm	77
4.13	Definition of Slip-Sinkage for Elastic Mode	78
4.14	Illustration of Re-Shearing Concept for Frictional Terrain	82
4.15	Comparison of Results Obtained Using Analytical and Numerical Procedures for Determining Re-Shearing Effects	85
4.16	Flowchart of Practical Tire-Terrain Interaction Model	87
4.17	Conversion of Normal Pressure and Shear Stress to Nodal Forces	90
4.18	Definitions of Shear Displacement and Slip-Sinkage for FEM Tire Model	95
4.19	Illustration of Negative Shear Displacement	99
4.20	Flowchart of Finite Element Tire-Terrain Interaction Model	102
4.21	Illustration of a Test Case for Verifying the Incremental-Iterative Scheme	105
5.1	Schematic of Forced-Slip Tire Tester	108
5.2	Free-Body Diagram of Forced-Slip Tire Tester	109
5.3	Block Diagram of Data Acquisition System for Forced-Slip Tire Test	112
5.4	Sample of Measured Forced-Slip Tire Test Data at 2% Slip	114
5.5	Sample of Measured Forced-Slip Tire Test Data at 5% Slip	115
5.6	Sample of Measured Forced-Slip Tire Test Data at 10% Slip	116
5.7	Sample of Measured Forced-Slip Tire Test Data at 20% Slip	117
5.8	Sample of Measured Forced-Slip Tire Test Data at 35% Slip	118

5.9	Sample of Measured Forced-Slip Tire Test Data at 50% Slip	119
5.10	Finite Element Tire Model for Simulating Tire-Terrain Interaction . .	127
5.11	Measured and Predicted Thrust at 30 psi (206.83 kPa)	129
5.12	Measured and Predicted Motion Resistance at 30 psi (206.83 kPa) . .	130
5.13	Measured and Predicted Drawbar Pull at 30 psi (206.83 kPa)	131
5.14	Measured and Predicted Total Sinkage at 30 psi (206.83 kPa)	132
5.15	Measured and Predicted Thrust at 20 psi (137.89 kPa)	133
5.16	Measured and Predicted Motion Resistance at 20 psi (137.89 kPa) . .	134
5.17	Measured and Predicted Drawbar Pull at 20 psi (137.89 kPa)	135
5.18	Measured and Predicted Total Sinkage at 20 psi (137.89 kPa)	136
5.19	Measured and Predicted Thrust at 10 psi (68.94 kPa)	137
5.20	Measured and Predicted Motion Resistance at 10 psi (68.94 kPa) . .	138
5.21	Measured and Predicted Drawbar Pull at 10 psi (68.94 kPa)	139
5.22	Measured and Predicted Total Sinkage at 10 psi (68.94 kPa)	140
5.23	Normal Pressure Distributions from Practical Model for Various Slips	150
5.24	Shear Stress Distributions from Practical Model for Various Slips . .	151
5.25	Illustration of Normal Pressure and Shear Stress Distributions from Practical Model for a Given Slip	152
5.26	Normal Pressure Distribution from FEM Model: Side View	154
5.27	Normal Pressure Distribution from FEM Model: Top View	155
5.28	Normal Pressure Distribution from FEM Model: Front View	156
5.29	Effect of Slip-Sinkage on Thrust for Practical Model	157
5.30	Effect of Slip-Sinkage on Motion Resistance for Practical Model . . .	158
5.31	Effect of Slip-Sinkage on Drawbar Pull for Practical Model	159
F.1	Variations of the Maximum Pressure/Weight and Shear Stress/Weight with Slip	297
F.2	Variation of the Position of the Maximum Pressure with Slip	298

Principal Symbols

A	contact area
A_0	terrain constant in unloading-reloading equation
A_s	terrain constant in slip-sinkage Equation (3.16)
A_u	terrain constant in unloading-reloading equation
B_s	terrain constant in slip-sinkage Equation (3.16)
DB	drawbar pull
E_t	Young's modulus for the tread of a tire
E_s	Young's modulus for the sidewall of a tire
F_c	cable force on forced-slip tire tester
F_t	towing force on forced-slip tire tester
F_p	bevameter penetration force
$\{F\}$	force vector
$\vec{F}_i$	force vector at iteration i
$\Delta\vec{F}_i$	difference in force vector at iteration i
$\vec{F}_k$	force vector at increment k
G_s	terrain constant in slip-sinkage Equation (3.18)
H	horizontal force in Bekker's slip-sinkage equation
H_s	terrain constant in slip-sinkage Equation (3.18)
K	shear deformation modulus
K_s	terrain constant in slip-sinkage Equation (3.16)
K_u	terrain constant in unloading-reloading equation
$[K]$	global stiffness matrix
M_s	terrain constant in slip-sinkage Equation (3.18)
R	non-slip rolling radius of pneumatic tire
R	radius of undeformed tire
R_1	radius of slip pulley
R_2	rolling radius of a tire

Principal Symbols

R_c	regression coefficient
R_m	motion resistance
T	thrust
T_q	bevameter torque
V_z	slip-sinkage velocity at wheel center
W	normal load applied
W_0	critical load
Z	sinkage at wheel center
Z_0	slip-sinkage at wheel center
b	width of a rectangular plate, or radius of a circular plate
c	cohesion
i	slip
j	shear displacement
k_c	cohesive modulus of terrain deformation
k_ϕ	frictional modulus of terrain deformation
n	exponent of terrain deformation
p	normal pressure
p_c	carcass pressure
p_g	ground pressure
p_l	lug pressure
p_n	nominal pressure
p_u	normal pressure at unloading point
r_i	inner radius of shear ring
r_o	outer radius of shear ring
r_p	radius of bevameter penetration plate

Principal Symbols

z	sinkage
z_j	slip-sinkage in Bekker's slip-sinkage equation
z_s	slip-sinkage, or predicted slip-sinkage
z_s^*	geometrically allowable slip-sinkage
z_{s0}	slip-sinkage at wheel center for a given θ
z_u	sinkage at unloading point
α	ratio of Young's moduli
α_r	tangential angle of a point on tire in contact area
γ	contact angle of a point in contact are after slip-sinkage
γ_0	contact angle corresponding to θ_0
γ_1	contact angle corresponding to θ_1
δ	tire deflection
$\{\delta\}$	displacement vector
ϵ	goodness of fit
ε	overall error in convergence criterion
η	local coordinate in finite element formulation
η_g	gear ratio
η_l	ratio of lug area over carcass area
θ	contact angle of a point in contact area
θ_0	initial contact angle
θ_1	final contact angle
θ_p	initial contact angle corresponding to angle θ
λ	scaling factor
ξ	local coordinate in finite element formulation
τ	shear stress
ϕ	angle of shearing resistance
ω	angular velocity of a tire

Chapter 1

Literature Review

The prediction and evaluation of the tractive performance of off-road wheeled vehicles is one of the major topics in Terramechanics. To predict their performance, it is essential to have a basic understanding of the mechanics of tire-terrain interaction. A variety of methods to the study of the mechanics of tires on unprepared terrain, ranging from entirely empirical to highly analytical, have been proposed. A brief review of these methods is presented below.

1.1 Review of Tire-Terrain Interaction Models

The tractive performance of a tire is usually defined by its thrust, motion resistance, sinkage, slip in relation to the applied normal load, torque and angular speed. Since the thrust, motion resistance and other parameters are determined by the normal and shear stress distributions along the tire-terrain interface, a central issue in the study of the mechanics of tire-terrain interaction is to establish the relationships between the stress distribution on the interface and tire parameters and terrain conditions [1].

Over the years, many methods have been proposed for determining the stress distributions on the tire-terrain interface. In 1960's Bekker [2] proposed a relatively simple method of evaluating the performance of a pneumatic tire on soil. It was assumed that if the sum of the inflation pressure and the pressure produced by the carcass is greater than or equal to the maximum pressure exerted by the terrain at

the lowest point of the tire circumference, the tire will remain undeformed like a rigid wheel. On the other hand, if the terrain is hard enough, the lower part of the tire will deform and the lower portion of the contact length will be equivalent to the chord of the circle representing the undeformed shape of the tire. The normal pressure along the contact patch is calculated from Bekker's pressure-sinkage relationship. The pressure varies from zero at the surface to a constant value along the flat portion of the contact patch. The resistance due to soil compaction, the internal resistance due to the hysteresis of tire material, and the sinkage are calculated from the normal stress distribution and equilibrium conditions. It should be mentioned, however, that the equations for evaluating the thrust and drawbar pull are not provided by this simple mathematical model developed by Bekker.

It should be pointed out that while the assumptions regarding tire deformation described above may be reasonable under certain conditions, they certainly can not be regarded as universally true.

In 1977, Fujimoto [3] proposed a tire performance model for operating over cohesive soil. A uniform normal pressure distribution along the contact patch was assumed. The shear stress was expressed as an exponential function of the shear strain of the soil. The contact geometry was considered as part of a rigid wheel with a radius greater than that of the undeformed tire. The front and rear contact angles were used in the analysis. The front contact angle was determined from the equilibrium of the tire, while the rear contact angle was expressed as a function of the ratio of tire stiffness and soil strength. The equivalent radius was calculated from the equilibrium equations. Making use of the equations for rigid wheel performance, the drawbar pull, motion resistance, torque and sinkage of the tire could be predicted.

This model appears to be an improvement over the Bekker model in that the shear stress on the tire-terrain interface has been taken into account. The torque, thrust and drawbar pull can then be determined from the shear stress distribution. However, the prediction procedure is more complicated than that of the Bekker model. Furthermore, the assumption that the normal pressure is uniformly distributed over the contact patch is not realistic in many cases.

Following Bekker's model, later Wong and Preston-Thomas [4] developed a comprehensive model for wheeled vehicle performance. It can be used to predict both

the normal and shear stress distributions under a tire. A new criterion for evaluating the operating mode of a tire, whether behaving like a rigid or elastic wheel, was introduced. In the criterion the ground pressure, p_g , which is defined as the ratio of the normal load applied on the tire to the projected contact area when the tire is on hard level ground, is compared with the nominal pressure, p_n , which is defined as the pressure at the lowest point of the tire in soft soil if it behaves like a rigid wheel. If the nominal pressure is equal to the ground pressure, then the tire is in the critical condition. Further increase in load will cause the tire to deform. Otherwise, it is in the rigid mode. Figure 1.1 illustrates the different operating modes of a tire. Effects of lugs on the tractive performance are taken into consideration.

In 1984, Baladi and Rohani [5] developed a model, in which the tire terrain interaction was treated as two springs in series, one representing the tire and the other representing the terrain. The stiffness of the tire spring was derived from the characteristics of the tire on hard ground. The contact length was assumed as the chord of a circle representing the tire circumference, and the contact width was assumed to remain constant. The rigidity of the terrain, on the other hand, was obtained by assuming that at certain sinkage the normal stress exerted by terrain is equal to the radial stress inside an expanding cavity. The radial stress was expressed analytically in terms of the shear strength and the volumetric change characteristics of soil. From these, the normal pressure on the interface was expressed as a function of contact angle and terrain properties. The shear stress was then derived in terms of normal pressure, slip, and the shear modulus, cohesion and angle of internal friction of the terrain. An assumption similar to Fujimoto's about the contact geometry was made. The performance was determined by integrating the normal and shear stresses over the contact area.

Baladi and Rohani showed that there was good agreement between the tire performance predicted using their model and the laboratory test data. It should be mentioned that according to this model the maximum normal pressure always occurs at mid-point of the contact patch. This does not agree with existing experimental data, which indicate that the location of the maximum normal pressure varies with the slip of the tire [6, 7].

Oida et al [7] measured the normal, longitudinal-tangential and lateral-tangential

Rigid Tire

$$p_g < p_n$$

Rigid-Elastic Transition

$$p_g = p_n$$

$$L_1 = 0$$

Elastic Tire

$$p_g = p_n$$

$$L_1 > 0$$

Figure 1.1 Operating Modes in Wong's Tire Model

stresses on the tire-soil contact surface using a forced-slip test apparatus and a so-called Krick Sensor, which can measure the three stress components noted above, in the laboratory. They found that the maximum pressure decreased with slip at a given load, the position of which moved forward with slip for slip below 65% and then moved backward for slip above 65%. They also found that the maximum longitudinal shear stress τ_{max} occurred always at a position ahead of the maximum normal stress σ_{max} and moved slightly backward with increasing slip. The value of the maximum longitudinal stress increased slightly with increasing slip.

More recently Wulfsohn and Upadhyaya [8, 9] developed two tire terrain interaction models, two-dimensional and three-dimensional, respectively, for predicting tractive performance and soil compaction. The three-dimensional contact geometry was measured using a special technique. This technique utilizes a transducer wire for measuring the lateral arc length changes at discrete locations along the contact length. The measured profile is then used to fit the soil deformation model to obtain the parameters in the model. To obtain the normal pressure and shear stress distributions in the contact region, several assumptions were made. First, it was assumed that all compaction within the soil volume is due entirely to the normal pressure. Second, a semi-logarithmic stress-volumetric strain relationship was assumed throughout the soil. Third, a strain function was assumed, the parameters of which were obtained by minimizing the sum of squared errors between the measured and predicted vertical loads. The tractive performance could then be predicted based on the predicted normal pressure and shear stress distributions. It should be pointed out that the tire deformation profile is required as input to the model.

The Finite Element Method (FEM) was first introduced into the study of tire terrain interaction by Perumpral et al [10] in 1971 in an attempt to predict the stress distributions and the soil deformation within the soil mass under a tractor tire. In this method, for a stationary tire the elliptical contact area was approximated by a circular area and the load was applied normal to the circular area uniformly. The non-linear behaviour of the soil was considered, and the non-linear equations were solved using an iterative method. The elastic constants for different elements may be different and were updated in each iteration according to the stress-strain relationship obtained from experimental data. The variation between the new and previous elastic

constants served as a criterion for convergence. The Poisson's ratio was assumed to be a constant throughout the analysis and a value of 0.4 was chosen for convenience. This means that the Young's modulus was the only variable. The results were compared with those from elastic theory and showed good agreement.

For a rolling tire, it was considered to be a plane strain problem. The tire was assumed as a rigid wheel to simplify the analysis. The assumptions regarding the elastic properties of soil were the same as those for a stationary tire. Sinlage, radial and tangential stresses were specified as inputs for the finite element analysis. The outputs were the strain, stress and displacement fields of the soil under the tire with a given load.

This kind of model requires the geometry of the contact patch and the stress distributions on the tire-soil interface to be specified at the outset. It can only be used to analyze the strain, stress and displacement fields within the soil mass. It can not be used to predict the stress distributions on the interface, the shape of the interface and consequently the tractive performance of the tire.

In 1977 and 1980, Yong et al [11] developed a more sophisticated finite element model. They assumed that the tire was a linear elastic cylindrical body (plane strain case). The terrain was treated as another cylindrical body with an infinite radius and as a piecewise linear elastic material. Only the terrain was discretized by finite elements. The contact length and deflection on hard surface were predicted using the Hertz contact theory for two elastic bodies, one of which representing the hard surface had a very high value of Young's modulus. To account for the tangential stress present on the contact surface the Hertz theory was modified. By matching the load-deflection characteristics on hard ground, the Young's modulus of the tire, which reflects the behaviour of the whole tire assembly, including the hub, tire carcass stiffness, and the inflation pressure, was obtained. The value of Poisson's ratio was assumed to be 0.5 for the reason that the volumetric change of the tire under load is usually insignificant. The normal and shear stress distributions on the contact surface were taken from experimental data and used as inputs to the finite element analysis. Again, this method is only suitable for studying the strain and stress fields in the soil mass. It should be pointed out that most unprepared terrain under load does not behave elastically, therefore, the underlying assumption for this model that the terrain

is piecewise elastic appears to be unrealistic. Compared with the model proposed by Perumpral et al [10], the model developed by Yong et al provides improvements in the predictions of contact geometry.

In 1984, Oida [12] developed an advanced finite element model for predicting the deformation of soil under tire loading without requiring the stress boundary conditions as input. In the model, the soil was discretized and considered as a rheological material, the characteristics of which are time-dependent, including creep and stress relaxation. Constitutive equations in both mono-axial and two-dimensional stress states were formulated. Time was a variable in his model. This implies that this model is non-linear. Constant strain or stress boundary conditions were assumed for a very short time in each calculation. The new boundary conditions for the subsequent calculation were from the previous FEM solution. In comparison with other finite element models developed so far, this model does not require the stress distributions on tire-terrain interface as inputs. It was claimed that this method could be applied to other analysis such as the prediction of soil deformation under a moving tire.

In 1988, Chen [13] studied the tire-terrain interaction using FEM. In his model the pneumatic tire was modeled as an elastic hollow cylinder with basic dimensions corresponding to those of the tire. Its characteristics were analyzed using FEM. In the analysis of tire-terrain interaction, semi-empirical relationships measured by the bevameter were used to model the terrain. Using an incremental numerical technique and considering the dynamic equilibrium conditions of a moving tire, the deformed shape of the tire, the sinkage, and the normal and shear stress distributions on the interface were determined. From the stress distributions, the tractive performance of a tire could be more realistically predicted. The procedure was then extended to the prediction of the overall performance of an off-road wheeled vehicle. However, the slip-sinkage effect was not considered in the model.

In most of the methods described above, the normal pressure across the contact width of a tire is implicitly assumed to be uniformly distributed. This is not always consistent with the available experimental data [6, 7]. The elastic rebound of the terrain at the trailing edge of the contact patch due to the unloading by the tire is not taken into consideration.

It is generally accepted that due to the nonlinearity of this type of problem, the ap-

plicability of the Hertz contact theory is very limited. Hence, finite element methods are widely used. Papers on the mechanics of contact of two bodies with rigid-elastic, elastic-elastic and even elastic-plastic characteristics and having simple contact geometry have been published. The problem is classified into two major groups, with friction and without friction on the contact surface. The methods presented in these papers [14, 15, 16] have been mainly based on variational principles and finite element techniques. While these methods could provide a fundamental understanding of the contact problem, they are only exemplified by a number of simple cases. As a result, they have not provided a realistic engineering solution to the tire-terrain interaction problem, with complicated contact geometry and complex response of the terrain to the normal and shear loadings.

An example of this kind of study was conducted by Oden and Lin [15] in 1986. A cylindrical deformable body which is rolling on a rough, flat hard surface was studied. The material of the cylinder was assumed to be a compressible, incompressible hyperelastic material or a hyperelastic / viscoelastic material characterized by constitutive equations valid for finite deformation. The friction on the contact surface, as well as the inertia of the cylinder, was taken into account. Bifurcations of the solutions of the model at certain rolling velocity were studied, which were manifested as the emergence of standing waves in the cylinder.

Another example was shown by Chen and Tsai [16] in 1985. The paper presented a general and effective finite element technique for analyzing elastodynamic sliding contact problems with friction. The deformed contact area was obtained using the constraint conditions developed by quadratic mathematic programming. Lagrangian multipliers were introduced to evaluate the contact pressure due to friction and to determine the adhesion or separation of contact surface. Based on these, the sliding process between two contact surfaces was accurately modeled. Their work also provided the correction formulae for modeling the transient response of velocities and accelerations on contact surface when initial impact or separation of the surface occurred. Several examples were given in the paper.

1.2 Review of Finite Element Tire Models on Hard Surface

In addition to the application of the finite element method to the analysis of contact problems mentioned above, FEM has also been employed for the analysis of tire characteristics and the performance of tire on paved roads.

In 1980 and 1981, the researchers at General Motors Research Laboratories [17, 18, 19] constructed a large scale, comprehensive, three-dimensional, thermal mechanical finite element model for analyzing the power loss in pneumatic tires rolling on hard ground. The model consists of three major analysis modules—deformation, dissipation, and thermal—which are used in an iterative and relaxation solution scheme.

In 1985, Ridha et al [20] constructed a three-dimensional finite element model of a homogeneous tire to study its deformation subject to a normal load on hard surface. The contact area was predicted using an incremental-iterative approach. For a given deflection of the hub, the program computes the nodes that come into contact with a pre-defined contact plane, and the magnitude of the contact forces at the nodes. It was found that the computed shape and size of the contact area at different levels of rim deflection were in good agreement with experimental results. The computed results included the distribution of stress, strain and strain energy within the tire and on the contact surface.

In 1988, Kenny and Stechschulte [21] developed a finite element tire model which uses a unique approach to obtain three elemental properties. The lamination model for materials of the tire carcass uses a numerical integration technique to determine the combined properties of all laminates within the element. In plane and out of plane couplings are predicted. The ability of the tire model to calculate interply shear strain is restricted only by the chosen shape function. Illustrations were given for inflated tire shape, interply shear strains, and cord loads in a typical bias mud and snow tire for light trucks.

A steady state formulation of the rolling contact problem with friction that allows the analysis of free rolling, cornering, acceleration, and braking was presented by Faria et al [22] in 1989. The formulation was applied to the finite element analysis

of tires. A layered shell finite element with shear deformation that allows for large deflection and rotation was developed. In each layer, orthotropic Hookean materials or Mooney-Rivlin type materials with fiber reinforcements can be used and the incompressibility constraint is enforced with the Lagrange multipliers. The contact constraint is enforced with a penalty. The friction term in the formulation, instead of the usual Coulomb friction, is regularized by a differentiable form that makes it more suitable for numerical analysis. A numerical example for a typical tire was also given.

In 1992, Faria et al [23] further enhanced their model, which includes the effects of spinning, cornering, contact, friction, pressurization, and applied torque for material bodies undergoing large displacements, rotation, and strains. The hyperelastic material characterization was extended to include a viscoelastic material model based on an internal state variable formulation. Several numerical examples demonstrating various aspects of the model were presented with qualitative agreement with experimental and theoretical results.

In 1993, Gall, Tkacik, and Andrews [24] built a three-dimensional finite element model for pneumatic tires with a deflected tread block structure to more accurately describe the stress distribution in the tire-ground contact area. Because of the importance of properly modeling the contact, several preliminary studies were detailed. First, the frictional contact between a rib and the ground was presented using a plane strain model. In particular, the edge effects were studied. Second, a combined experimental and numerical method for determining the coefficient of friction between a rubber block and a glass plate was described. A friction law was developed which includes the well-known dependence of the coefficient of friction on the normal stress. Finally, the three-dimensional model was discussed and results were given.

Nakashima and Wong [25] developed a three-dimensional finite element tire model. They further improved Wang's tire model [26] by introducing two equivalent Young's moduli into their model. One represents the Young's modulus of the tire tread, and the other represents that of the sidewall. The values of Young's moduli were deduced from the contact area-deflection and load-deflection relationships of the tire. These relationships were either from direct measurements or from the "Generalized Deflection Chart" provided by tire manufacturers. A numerical scheme for searching

for the ratio of Young's moduli was developed. Numerical examples were provided for validating the tire model.

Most of these research efforts [14, 17, 18, 19, 27] were concentrated on the analysis of a tire under vertical loading on hard ground for a given inflation pressure. The analysis of pneumatic tire performance over deformable terrain was not included in these studies. However, their modelling methods and formulations provide a basis for the development of models for tires interacting with soft terrain.

1.3 Review of Studies of Slip-Sinkage

When a torque is applied to a tire, a thrust will be developed on the tire-terrain interface. This results in tire slip. The slip of the tire causes an additional sinkage, which is usually referred to as "slip-sinkage". While slip-sinkage phenomenon has long been observed, the mechanics behind such phenomenon has yet to be elucidated.

Bekker [28] proposed a method for estimating slip-sinkage. It was assumed that there exists a similarity between the force vector diagram and the velocity vector diagram for a cleat as illustrated in Fig. 1.2. From the similarity the following relation can be established,

$$\tan \alpha_b = \frac{j}{z_j} = \frac{H}{W - W_o} \quad (1.1)$$

where j is the shear displacement, z_j is the slip-sinkage, W is the normal load applied, W_o is the bearing capacity of the soil, H is the horizontal force, and α_b is the angle between the resultant displacement and the vertical.

Therefore, the slip-sinkage can be calculated as follows,

$$z_j = \frac{j(W - W_o)}{H} \quad (1.2)$$

It should be noted that Eq. (1.2) only provides a method for calculating slip-sinkage. It does not give any insight into the mechanics involved.

In 1965, Reece [29] proposed a wedge theory in an effort to explain the slip-sinkage phenomenon. It was assumed that a soil wedge is formed under a sliding plate. Due to the horizontal movement of the plate, part of the shear strength on the side of

Figure 1.2 Bekker's Slip-Sinkage Model

the soil wedge is used to balance the horizontal force. Consequently, the original equilibrium can not be maintained. To restore equilibrium, the normal force on the sides of the soil wedge must be altered, and considerable additional sinkage, which is slip-sinkage, is required to balance the applied vertical load.

Figure 1.3 illustrates the Reece theory. The original equilibrium state is shown in Fig. 1.3a, where w is the load applied, ϕ is the angle of internal friction and p is the normal force on the sides of the soil wedge, which can be determined from the original equilibrium condition. The second equilibrium state is shown in Fig. 1.3b, where p' is the altered normal force on the sides of the soil wedge, which can be determined from the second equilibrium condition.

Unlike Bekker's method the Reece theory provides certain insight into the mechanics of slip-sinkage. However, the slip-sinkage of a plate for a given loading condition can not be quantitatively determined from the Reece theory.

1.4 Scope of the Present Work

The literature review presented above indicates that there is a need for developing more accurate and realistic numerical models for predicting the tractive performance of pneumatic tires over soft terrain, particularly taking into account the slip-sinkage effect. This thesis is intended to develop two new numerical models for simulating the interactions between pneumatic tires and soft terrain, to predict the tractive performance of tires using the models and to validate the models experimentally. Both semi-empirical and finite element techniques will be used for developing the numerical simulation models.

One of the numerical models is based on semi-empirical approach, and its formulation is relatively simple in comparison to the finite element technique. Therefore, it is hereinafter referred to as the "practical model". The other is a three-dimensional finite element model, which is more realistic in terms of tire contact geometry but demands much more computing power to obtain the deformed shape of the tire. In both models, the terrain is represented by a group of semi-empirical equations describing the pressure-sinkage relationship, the response of terrain to repetitive loadings, the shear stress-shear displacement relationship, and the slip sinkage shear displacement

Figure 1.3 Reece's Slip-Sinkage Theory

relationship. Different approaches for incorporating slip-sinkage into tire terrain interaction models are also investigated.

The practical model utilizes the tire model developed by Wong [4] and his criterion for determining the operating mode of a tire over deformable terrain. Slip-sinkage effect is taken into account. The equilibrium conditions and the work equivalence relationship are used to determine the sinkage and slip-sinkage at the wheel center. The normal pressure and shear stress distributions can then be determined, and the tractive performance of the tire can be evaluated.

The finite element tire model follows that developed by Nakashima and Wong [25], in which the tire is idealized as a three-dimensional structure of smoothed exterior surface with two equivalent Young's moduli, one for the sidewall and the other for the tread, because of their apparent differences in stiffness. The values of Young's moduli are obtained based on the relationship between the contact area and deflection and that between the load and deflection. These two relationships are derived from the test data measured using the static tire tester. Slip-sinkage is incorporated as for the practical model. An incremental-iterative method is developed for obtaining vertical equilibrium. Both models make use of the tire characteristics obtained from the static tire tests, or from the "Generalized Deflection Chart" provided by the tire manufacturer, and of the terrain properties measured using the bevameter technique, as inputs. They can be used to predict the tractive performance of tire over deformable terrain for a given slip under a specific loading condition.

Experimental validation of the models are carried out using the full-scale forced-slip tire tester available at the Transport Technology Research Laboratory, Carleton University. The measured and predicted tractive performances, including thrust, motion resistance, drawbar pull and sinkage, are compared. Conclusions of the study and recommendations for future work are presented.

Chapter 2

Finite Element Model of Pneumatic Tires

This chapter describes the construction of the finite element tire model, the static tire testing, and the method for deriving the equivalent material properties of the finite element tire model from the static tire test data. Numerical results of the tire model on hard ground are compared with the measurements from static tire tests to verify the model.

2.1 Finite Element Representation of Tires

The necessity of using the finite element technique in modeling pneumatic tires has been discussed by many researchers [13, 25, 26]. It is mainly used for modeling the contact geometry more realistically for both tire on hard ground and on soft terrain. In this thesis the method developed by Nakashima and Wong [25] is employed.

The finite element tire model is illustrated in Fig. 2.1. It can be seen that the tire model takes the major geometric dimensions of an actual tire into account except the lugs. This is because it is very difficult to express the normal pressure and shear stress around lugs using semi-empirical equations when modelling tire-terrain interaction. It is, therefore, assumed that space between the lugs of the tire is filled with soil when operating over terrain, and all forces are acting on the outer surface of the tire.

An actual pneumatic tire carcass consists of rubber with high strength cords em-

Total Nodes: 660 Total Elements: 294

**Transport Technology Research Laboratory
Carleton University, Ottawa, CANADA**

Figure 2.1 Finite Element Tire Model for Simulating Static Test

bedded, steel beads along the rim and various kinds of lugs on the outside of carcass. From the viewpoint of the study of tire-terrain interaction, the internal structure of the tire carcass is not the major concern, rather the mechanical behaviour and the tread of the tire, which is in contact with the terrain, are of greater importance. Therefore, the complex material properties of the tire carcass are lumped into two equivalent Young's moduli, one for the tread and other for the sidewall due to their apparent difference in stiffness. The elastic properties of the tread are represented by an equivalent Young's modulus E_t , and that of the sidewall is represented by E_s . The equivalent Poisson's ratio is assumed to be close to 0.5 based on the observation that the volumetric change in carcass is insignificant. The stiffness difference between the tread and the sidewall can be characterized by the ratio $\alpha = E_s/E_t$. Although the detailed structure of lugs is ignored in the tire model, when analyzing tire-terrain interaction using this model, the influence of lugs on the tractive performance is taken into account to a certain extent. This will be discussed in Chapter 4.

For setting up a coordinate system for the finite element model of the tire, the origin of the system is located at the wheel center with y -axis in the vertical direction, x -axis parallel to level hard ground and z -axis in the axial direction (see Fig. 2.1). Since all loads are symmetric about x - y and y - z planes, only a quarter of the whole tire is modeled. The 8-node isoparametric three-dimensional element is selected for reasons of simplicity. Parametric studies of tire mesh indicated that using a total number of nodes of 660 and a total number of elements of 294 for the model is adequate.

2.2 Derivation of the Young's Moduli

In the tire industry, the finite element method has been widely used in the development and design of new products, and sophisticated computer models have been developed. However, these models generally require detailed information, as input, on the mechanical properties of rubber compounds, cords, belts, leads, etc. This kind of information is usually of a proprietary nature and is not available to the tire user. On the other hand, it would not be cost-effective for the user to directly measure

the mechanical properties of these components. Therefore, a system identification approach is followed to derive the equivalent elastic properties of the tread and of the sidewall from the load-deflection relationship and contact area deflection relationship. Such relationships are illustrated in Figs. 2.5 and 2.6, which are from static tire tests conducted in the laboratory.

It is well known that in finite element analysis the elastic system in question is represented by the stiffness matrix $[K]$, force vector $\{F\}$ and displacement vector $\{\delta\}$ as follows [30, 31],

$$[K]\{\delta\} = \{F\} \quad (2.1)$$

For nodes in the region of a single isotropic material, such as the tire tread, the equations involve only the properties of that material,

$$\sum_{j=1}^n a_{ij} E_t \delta_j = F_i \quad (2.2)$$

or

$$\sum_{j=1}^n a_{ij} \delta_j = \frac{F_i}{E_t} \quad (2.3)$$

where a_{ij} is a coefficient involving geometry and Poisson's ratio, E_t is the Young's modulus, δ_j is the displacement, and F_i is the nodal force.

Equation (2.3) shows that if δ_j is given, the quantity of $\frac{F_i}{E_t}$ is treated as an unknown and can be determined without knowing the actual value of E_t . On the other hand, if δ_j is unknown, then F_i must be given, which is zero in the case of tire hard ground interaction. This provides a means for determining the deformation of the body in question without knowing the Young's modulus E_t , provided the inputs to the body are displacements only.

Similarly, for nodes connecting two isotropic materials, such as those between the tire tread and the sidewall, the coefficients for this node involve the properties of both materials,

$$\sum_{j=1}^n [(a_{ij})_1 E_t + (a_{ij})_2 E_s] \delta_j = F_i \quad (2.4)$$

or

$$\sum_{j=1}^n [(a_{ij})_1 + (a_{ij})_2 \alpha] \delta_j = \frac{F_i}{E_t} \quad (2.5)$$

where $(a_{ij})_1$ is a coefficient from the first material, $(a_{ij})_2$ is from the second, α is the ratio of E_s to E_t , and others are the same as for Eq. (2.3).

If the prescribed displacements in Eq. (2.5) are given and the value of α is known, then the deformation of the tire can be determined without knowing the value of E_t , based on the same reasoning for the case of one Young's modulus.

From the above analysis, it is clear that in deriving the elastic properties for the tire model, the ratio α can be searched first using the relationship between contact area and deflection (or contact length-deflection, or contact width-deflection relationships). The values of the Young's moduli, E_t and E_s , can then be quantitatively determined using the load-deflection relationship.

Figure 2.2 depicts the method for searching for the values of α . To initiate the search two trial values for α_1 and α_2 ($\alpha_1 < \alpha_2$) are assumed. Then two sets of relationships between tire deflection δ and contact area A can be obtained by using the tire model with prescribed displacement conditions, as shown in Fig. 2.2. Strictly speaking, a relationship between α and δ is needed, such that when this relationship is used as input to the tire model, a contact area-deflection curve that is identical to the measured one can be produced from the tire model.

To obtain such a relationship, first a linear relationship between the two derived $A-\delta$ (contact area-deflection) curves and the measured $A-\delta$ curve is established. Then a point on the measured curve, δ^k and A^k as shown in Fig. 2.2, can be determined for given points on the two derived curves using the following relations,

$$A = c\delta + d \quad (2.6)$$

where

$$c = \frac{A_1^k - A_2^k}{\delta_1^k - \delta_2^k}$$

$$d = \frac{A_1^k \delta_2^k - A_2^k \delta_1^k}{\delta_1^k - \delta_2^k}$$

Figure 2.2 Method for Searching for Young's Modulus Ratio α

and A_1^k and A_2^k are the contact area from the tire model at deflections δ_1^k and δ_2^k for α_1 and α_2 , respectively.

Second, it is assumed that a linear relationship between α and δ exists among the A - δ curves,

$$\alpha^k = \frac{\delta_1^k - \delta_2^k}{\delta_1^k - \delta_2^k}(\alpha_2 - \alpha_1) + \alpha_1 \quad (2.7)$$

From Eq. (2.7) various values of α^k can be obtained for a given A - δ curve. The relationship between α and δ can then be obtained by fitting those values of α with a polynomial. The method described above is considered as one of the simplest methods for deriving the relationship between α and δ . Other methods may also be devised for determining the α - δ relationship.

In the method described above, the use of the contact area-deflection relationship for determining α is presented. Other geometric relationships, such as contact length-deflection and contact width-deflection, can also be used. If all three relationships are used for determining the α - δ relationship, the accuracy of the derived relationship may be improved.

After the α - δ relationship has been determined, it is then possible to determine the values of Young's modulus E_t or E_s based on the measured load-deflection relationship of the tire. If the value of E_t is to be searched, then the value of E_s can be determined from the ratio α , and vice versa.

The search method developed by Nakashima and Wong [25] for determining the values of Young's moduli is employed. To initiate the search, an initial value of E_t is assumed. Using this E_t and a value of α from the α - δ relationship derived, the normal reaction of the tire model at the corresponding δ can be calculated. It is then compared with the normal load applied. If the difference between the normal reaction from the ground and the normal load applied to the tire is not within a prescribed error, then the value of E_t is adjusted and the procedure is repeated until the error is less than the prescribed error.

Later studies show that the values of Young's moduli may also be directly solved without incorporating any search scheme. This can be seen from Eq. (2.3) and

Eq. (2.5). Once the displacements are prescribed and the value of α is given, the left-hand sides of both equations become constants. Take Eq. (2.5) for example,

$$K_i = \sum_{j=1}^n [(a_{ij})_1 + (a_{ij})_2 \alpha] \delta_j = \frac{F_i}{E_t} \quad (2.8)$$

It is clear from Eq. (2.8) that E_t and F_i are linearly related. The total reaction in the vertical direction is then,

$$W = \sum_{i=1}^m (F_i)_y \quad (2.9)$$

where W is the normal load applied and $(F_i)_y$ is the force component in the vertical direction.

Substituting Eq. (2.8) into Eq. (2.9), one obtains,

$$\begin{aligned} W &= \sum_{i=1}^m (K_i)_y E_t \\ &= E_t \sum_{i=1}^m (K_i)_y \end{aligned} \quad (2.10)$$

where $(K_i)_y$ is the coefficient associated with $(F_i)_y$.

Then the value of Young's modulus E_t for a given δ is,

$$E_t = \frac{W}{\sum_{i=1}^m (K_i)_y} \quad (2.11)$$

Equation (2.11) indicates that the value of Young's modulus E_t may be obtained in a more efficient way. It is recommended that this method be implemented in the future to improve the efficiency of deriving the equivalent Young's moduli for the tire model.

2.3 Static Testing of Pneumatic Tires

The purpose of the static test of a pneumatic tire is to obtain the relationship between the normal load and deflection, and that between the contact area and deflection for various inflation pressures. These two relationships are used for deriving the

Figure 2.3 Schematic of Static Tire Tester

equivalent Young's moduli for the finite element tire model, and for determining the operating mode of the practical tire model, which will be discussed later.

The static tire tester available at Carleton University is illustrated in Fig. 2.3. As can be seen, the tester consists of a mount, hydraulic jack, platform and tire shaft assembly. The hydraulic jack is fixed onto the middle portion of the mount. On the top of the jack is the load cell, on which the platform is placed. The tire shaft assembly is fixed onto the top of the mount. In addition a potentiometer is attached to the mount for monitoring the tire deflection, as shown in Fig. 2.3.

During the test, the portion of the tire carcass contacting the platform is sprayed with black ink or chalk powder and a piece of paper with grids is placed on the platform. Then the platform is raised by pumping the hydraulic jack. As the platform pushes against the tire, the tire will deform and a footprint of the tire will be left on the paper, which is considered as the projected contact area. The tire deformation and the load applied are recorded using a computerized data acquisition system as outlined in Fig. 2.4.

The tire used in the static tire test was a Michelin 11.00R16 XL. Its major dimensions are shown in the following table,

Table 2.1: Michelin 11.00R16 XL Tire Dimensions

	Michelin 11.00R16 XL
Unloaded Overall Diameter, mm	975
Unloaded Section Width, mm	279
Tread Depth, mm	45
Loaded Radius, mm	452

Tests for different inflation pressures were conducted. The results, both the relationship between load and deflection and that between contact area and deflection, are shown in Figs. 2.5 and 2.6, respectively. It can be seen from the test results that the inflation pressure affects the load-deflection curve as indicated in Fig. 2.5 and that it does not affect significantly the contact area-deflection relationship as shown in Fig. 2.6. It is, therefore, assumed that the relationship between contact area

Figure 2.4 Block Diagram of Data Acquisition System for Static Tire Test

Michelin 11.00R16 XL on Hard Ground

Figure 2.5 Measured Load-Deflection Data

and deflection is independent of inflation pressure in deriving the equivalent Young's moduli for the finite element tire model.

2.4 Verification of Finite Element Tire Model

The computer simulation codes developed by Nakashima and Wong [25] were used to simulate the tire-hard ground interaction using the tire model. The maximum deflection for the simulation was set at 9 cm. The simulations were carried out on Sun Sparc II workstations available in the Department of Mechanical and Aerospace Engineering, Carleton University. Numerical results are illustrated in the following figures.

Figures 2.7, 2.8 and 2.9 show the variations of the calculated and fitted values of Young's moduli, E_t and E_s , with normal load for inflation pressures 30 psi (206.83 kPa), 20 psi (137.89 kPa) and 10 psi (68.94 kPa), respectively. It can be seen from the figures that value of E_t is less than that of E_s at a given normal load for a particular inflation pressure, which may imply that the sidewall is stiffer than the tread. It is worth mentioning that the Young's moduli derived are for the carcass under a given inflation pressure and not the carcass alone.

It can also be seen that the Young's modulus for the tread, E_t , approaches a more or less constant value as the normal load increases. On the other hand, the Young's modulus for the sidewall, E_s , varies in a wide range. Therefore, it is necessary to use a special numerical technique to handle such finite element problems with material nonlinearities, when analyzing the interaction between the tire and deformable terrain.

Figure 2.10 shows three load-deflection curves for three different inflation pressures, it can be seen that the load-deflection curves from the model agree very well with the measurements as expected. The contact area-deflection curve as shown in Fig. 2.11 agree reasonably well with the measurements. Discrepancies shown in Fig. 2.11 are due to approximating the contact area with a rectangle and the method for searching for the ratio of Young's moduli. Nonetheless, a satisfactory numerical model for a pneumatic tire has been established.

Figure 2.7 Derived Young's Moduli for 30 psi (206.83 kPa)
Inflation Pressure

Figure 2.8 Derived Young's Moduli for 20 psi (137.89 kPa)
Inflation Pressure

Figure 2.9 Derived Young's Moduli for 10 psi (68.94 kPa)
Inflation Pressure

Michelin 11.00R16 XL on Hard Ground

Figure 2.10 Measured and Predicted
Load-Deflection Curves

**Figure 2.11 Measured and Predicted
Contact Area-Deflection Curves**

Chapter 3

Semi-Empirical Model of Terrain

In this chapter the bevameter technique, and the methodologies for processing the test data measured using the bevameter and for characterizing terrain behaviour are presented.

3.1 Bevameter Tests

In cross-country operations, a variety of terrains, ranging from desert sand through deep mud to snow, may be encountered. Their properties have a significant influence on the performance of off-road vehicles. In this thesis, the behaviour of the terrain is modelled using a group of semi-empirical equations.

Over the years, many methods have been proposed for characterizing terrain behaviour. Among them, the bevameter proposed by Bekker is one of the most widely used. It is based on the premise that the terrain properties pertinent to vehicle mobility can best be measured under the loading condition similar to that exerted by an off-road vehicle. A vehicle exerts normal and shear loads on the terrain surface. To simulate these, the bevameter technique comprises of two separate sets of tests: one is the pressure-sinkage test and the other is the shear test. The results of pressure-sinkage test would allow vehicle sinkage and motion resistance due to terrain compaction to be predicted, while those of the shear test would provide a basis for

predicting vehicle traction. The literature survey shows that the bevameter has been used for characterizing a variety of terrain, such as sand, clayey soil, muskeg and snow [32, 33, 34]. Good predictions of tractive performance were achieved using the terrain parameters obtained using the bevameter [4, 35, 36, 37]. Thus, the bevameter is selected to measure the properties of the terrain used in this study.

The schematic of the bevameter is shown in Fig. 3.1. It consists of the pressure-sinkage test component, the shear test component and the hydraulic control unit. In the pressure-sinkage test, the sinkage plate is attached to a shaft, which is connected with the piston of the hydraulic unit through the load cell. In the shear test, the shear head is driven by the same hydraulic unit through a chain mechanism. A torque cell is used to monitor the torque acting on the shear ring. Two potentiometers are used to monitor the sinkage, shear displacement and slip-sinkage.

The pressure-sinkage test and shear test were performed separately. The test data were collected using the computerized data acquisition system as outlined in Fig. 3.2. The methods for processing the test data are described in the following section.

3.1.1 Pressure-Sinkage Test

In the pressure-sinkage test, different sizes of plates are used, and both the penetration force and the sinkage are recorded. The penetration speed is usually kept low, at approximately 1 cm/s, to reduce dynamic effect. The pressure on the sinkage plate is obtained by dividing the penetration force by the contact area of the plate,

$$p = \frac{F_p}{\pi r_p^2} \quad (3.1)$$

where p is the normal pressure, F_p is the penetration force and r_p of the radius of the plate.

Sample pressure-sinkage data obtained in dry sand with two plates of diameters of 7.5 cm and 10 cm are shown in Fig. 3.3. It is seen that the two sets of curves are very close to each other, which means the size effect is very small.

Figure 3.1 Schematic of Bevameter

Figure 3.2 Block Diagram of Data Acquisition System for Bevameter Test

Bevameter Pressure-Sinkage Tests in Sand

Figure 3.3 Sample of Measured Pressure-Sinkage Data

3.1.2 Torque–Shear Displacement Test

In the shear test, two different types of shear rings are used. One is for measuring the internal shear strength, which is a ring with grousers, the other is for measuring rubber-terrain shearing characteristics, which is a ring covered with natural rubber. For each type of test, various normal loads are applied on the shear ring. The torque, shear displacement (rotation of the shear ring) and slip-sinkage (additional sinkage) are recorded for each test. The shear stress is obtained by dividing the torque by the product of the contact area of the shear ring and the average radius of the shear ring, i.e.,

$$\tau = \frac{T_q}{\frac{\pi}{2}(r_o + r_i)(r_o^2 - r_i^2)} \quad (3.2)$$

where τ is the shear stress, T_q is the torque measured, r_o and r_i are the outer and inner radii of the shear ring, respectively.

Sample shear stress–shear displacement data are shown in Figs. 3.4 to 3.7. Figures 3.4 and 3.5 show the variations of shear stress and slip-sinkage with shear displacement under different normal loads for the internal shearing of the terrain. Figures. 3.6 and 3.7 show similar results for rubber-terrain shearing.

3.2 Methods for Processing Bevameter Data

In this section, the methods for processing pressure–sinkage test data, shear stress–shear displacement data and slip sinkage–shear displacement data are described.

3.2.1 Pressure–Sinkage Relationship

The pressure–sinkage relationship used in this thesis is the equation proposed by Bekker [38], which takes the following form,

$$p = \left(\frac{k_c}{b} + k_\phi\right)z^n \quad (3.3)$$

where p is the pressure, z is the sinkage, b is the contact width of a rectangular plate or the radius of a circular plate, and k_c , k_ϕ and n are terrain constants.

Bevameter Internal Shear Tests in Sand

Figure 3.4 Sample of Measured Internal
Shear Stress-Shear Displacement data

Bevometer Internal Shear Tests in Sand

Figure 3.5 Sample of Measured Internal Slip Sinkage-Shear Displacement Data

Bevameter Rubber Shear Tests in Sand

Figure 3.6 Sample of Measured Rubber
Shear Stress-Shear Displacement Data

Bevameter Rubber Shear Tests in Sand

Figure 3.7 Sample of Measured Rubber
Slip Sinkage-Shear Displacement Data

Equation (3.3) requires at least two tests with different sizes of plates to obtain the three constants. A weighted least squares method is used to derive those parameters. The weighted error function F is given by [39],

$$F = \sum p_m^2 [\ln p_m - \ln k_{eq} - n \ln z]^2 \quad (3.4)$$

where p_m is the measured pressure, and $k_{eq} = \frac{k_c}{b} + k_\phi$.

Taking partial derivatives of F with respect to n and k_{eq} , setting them to zero and solving for n and k_{eq} yield the following expressions,

$$n = \frac{(\sum p_m^2)(\sum p_m^2 \ln p_m \ln z) - (\sum p_m^2 \ln p_m)(\sum p_m^2 \ln z)}{(\sum p_m^2)(\sum p_m^2 (\ln z)^2) - (\sum p_m^2 \ln z)^2} \quad (3.5)$$

$$\ln k_{eq} = \frac{\sum p_m^2 \ln p_m - n \sum p_m^2 \ln z}{\sum p_m^2} \quad (3.6)$$

In practice it is likely for two tests with different sizes of plates to have different values of n . Since Eq. (3.3) implies that for a given terrain condition the value of n should be the same, the average value of n is used,

$$n_{av} = \frac{n_1 + n_2}{2} \quad (3.7)$$

Using n_{av} the value of k_{eq} is re-calculated,

$$\ln k_{eq} = \frac{\sum p_m^2 \ln p_m - n_{av} \sum p_m^2 \ln z}{\sum p_m^2} \quad (3.8)$$

The values of k_c and k_ϕ are calculated from the two values of k_{eq} obtained using the two sizes of plates,

$$k_c = \frac{k_{eq1} - k_{eq2} b_1 b_2}{b_2 - b_1} \quad (3.9)$$

$$k_\phi = k_{eq1} - \frac{k_{eq1} - k_{eq2} b_2}{b_2 - b_1} b_1 \quad (3.10)$$

where the subscripts, 1 and 2, are associated with the sizes of the plates.

The pressure-sinkage parameters obtained from the test data are shown in Table 3.1, and will be used as inputs to the tire-terrain interaction models.

Table 3.1: Pressure-sinkage Parameters

k_c (kPa/m ⁿ⁺¹)	k_ϕ (kPa/m ⁿ⁺²)	n	K_0 (kPa/m)	A_u (kPa/m ²)
29.31	8378.5	1.68	0.0	503000

The parameters K_0 and A_u in Table 3.1 are used in the equation for characterizing the terrain repetitive loading characteristics (see Section 4.1.1 for details).

Figure 3.3 shows a comparison of the measured data and fitted curves using the pressure-sinkage parameters shown in Table 3.1.

3.2.2 Shear Stress-Shear Displacement Relationship

The shear stress-shear displacement relationship used in this thesis is the exponential equation commonly used [8, 9, 35],

$$\tau = \tau_{max}(1 - e^{-\frac{j}{K}}) \quad (3.11)$$

where τ is the shear stress, τ_{max} is the maximum shear strength of the terrain, j is the shear displacement, and K is the shear deformation modulus, which is a parameter defining the magnitude of the shear displacement required to develop the maximum shear stress.

To obtain the shear parameters, τ_{max} and K , the following function based on the weighted least squares principle is used [39],

$$F = \sum \left(1 - \frac{\tau_m}{\tau_{max}}\right)^2 \left[\ln\left(1 - \frac{\tau_m}{\tau_{max}}\right) + \frac{j}{K} \right]^2 \quad (3.12)$$

where $\left(1 - \frac{\tau_m}{\tau_{max}}\right)^2$ is the weighting factor, τ_m is the measured shear stress.

Taking the partial derivative of function F with respect to K , setting it to zero and solving for K yield the following expression,

$$K = -\frac{\sum w_i j^2}{\sum w_i j \ln\left(1 - \frac{\tau_m}{\tau_{max}}\right)} \quad (3.13)$$

where $w_i = \left(1 - \frac{\tau_m}{\tau_{max}}\right)^2$.

Although the above expression gives the value of K that minimizes the error in curve fitting, the value of τ_{max} is still unknown. To obtain the value of τ_{max} , an iterative procedure has to be employed, as outlined below:

1. estimate the τ_{max} from the given data set.
2. calculate K using Eq. (3.13).
3. evaluate the goodness of fit ϵ using the following definition [39],

$$\epsilon = 1 - \frac{\sqrt{\frac{\sum (\tau_m - \tau_c)^2}{N-2}}}{\frac{\sum \tau_m}{N}} \quad (3.14)$$

where τ_c is the shear stress calculated using Eq. (3.11).

4. check whether the goodness of fit has reached its maximum. If yes, the τ_{max} has been found, and the process is then terminated. If not, the τ_{max} is estimated based on the current information, and the process is repeated.

The maximum shear strength is assumed to observe the Mohr-Coulomb failure criterion,

$$\tau_{max} = c + p \tan \phi \quad (3.15)$$

where c is the cohesion, p is the normal pressure applied, and ϕ is the angle of shearing resistance.

The parameters, c and ϕ , can be obtained from the maximum shear stresses derived using the above procedure for various normal pressures. The shear parameters for both internal and rubber-terrain shearing obtained from the test data are shown in Table 3.2, and will be used as inputs to the tire terrain interaction models.

Table 3.2: Shear Parameters

Shear Type	cohesion, c (kPa)	shearing Angle, ϕ (deg)	Shear Deformation Modulus, K (cm)
Internal	1.08	25	1.9
Rubber	0.22	27.7	1.9

Figures 3.4 and 3.6 show a comparison of the measured data and fitted curves using the shear parameters shown in Table 3.2.

3.2.3 Slip Sinkage–Shear Displacement Relationship

To characterize the slip sinkage–shear displacement relationship, the following relationship is proposed, based on experimental data,

$$z_s = (A_s + pB_s)(1 - e^{-\frac{j}{K_s}}) \quad (3.16)$$

where z_s is the slip-sinkage, p is the normal pressure applied, j is the shear displacement, and A_s , B_s and K_s are constants.

Equation (3.16) has the same form as Eq. (3.11). However, from the slip-sinkage test data shown in Figs. 3.5 and 3.7, it can be seen that the maximum slip-sinkage or the maximum value of $A_s + pB_s$ for a given p can not be readily identified. Therefore, a general method for fitting experimental data with nonlinear expression is used. A brief description of this algorithm [40] is provided in Appendix A.

It should be pointed out that Eq. (3.16) is used mainly for fitting the experimental data. The physical significance of the coefficients, A_s , B_s and K_s , are not apparent. In his study of slip-sinkage phenomenon, Reece [29] conceived that the slip-sinkage is caused by two factors: frictional shearing and cohesive shearing. It appears that coefficient A_s is associated with the cohesive shearing and B_s is associated with the frictional shearing.

The slip-sinkage parameters for both internal and rubber–terrain shearing obtained from the shear test data are shown in Table 3.3, and will be used as inputs to the tire–terrain interaction models.

Table 3.3: Slip-Sinkage Parameters for Eq. (3.16)

Slip-Sinkage Type	A_s (m)	B_s (m/kPa)	K_s (m)
Internal	1.55E-2	2.68E-3	0.14
Rubber	-3.5E-3	3.23E-3	0.11

Figures 3.5 and 3.7 show a comparison of the measured data and fitted curves using the slip-sinkage parameters shown in Table 3.3

Although Eq. (3.16) fits the experimental data quite well, it was found that the expression fails to describe the physical process of slip-sinkage adequately. Taking the second derivative of Eq. (3.16) with respect to time, one can obtain,

$$\frac{d^2 z_s}{dt^2} = -\frac{A_s + pB_s}{(K_s)^2} \left(\frac{dj}{dt}\right)^2 e^{-\frac{t}{K_s}} \quad (3.17)$$

where $\frac{d^2 z_s}{dt^2}$ represents the vertical acceleration of the shear ring, $\frac{dj}{dt}$ is the horizontal shear velocity, which is considered a constant.

From Eq. (3.17) it can be seen that the vertical acceleration is always negative. This is contradictory to the fact that since the shear ring or plate starts moving from rest, there must be some initial positive acceleration, although it may be very small.

In an effort to explain the physical process of slip-sinkage and yet fit the experimental data, another form of relationship for slip-sinkage has been examined, which is presented below,

$$z_s = \frac{G_s p^{H_s} j^2}{M_s + j^2} \quad (3.18)$$

where G_s , H_s and M_s are constants, z_s , j and p are defined as before.

It is not difficult to see that z_s of Eq. (3.18) is zero, when j is zero. When j approaches infinity, z_s stabilizes at $G_s p^{H_s}$. Furthermore, taking the second derivative of Eq. (3.18) with respect to time, one obtains the vertical acceleration,

$$\frac{d^2 z_s}{dt^2} = \frac{2G_s p^{H_s} M_s (M_s - 3j^2)}{(M_s + j^2)^3} \left(\frac{dj}{dt}\right)^2 \quad (3.19)$$

From Eq. (3.19) it can be seen that the vertical acceleration is zero at j equal to $\sqrt{\frac{M_s}{3}}$. When j is less than $\sqrt{\frac{M_s}{3}}$, the acceleration is positive, when j is greater than $\sqrt{\frac{M_s}{3}}$, the acceleration is negative. Therefore, this expression describes the physical process of slip-sinkage more closely.

The slip-sinkage parameters obtained from fitting Eq. (3.19) with experimental data are tabulated in Table 3.4.

Table 3.4: Slip-Sinkage Parameters for Eq. (3.18)

Slip-Sinkage Type	G_s (m)	H_s	M_s (m ²)
Internal	1.79E-03	1.17	5.23E-03
Rubber	8.64E-03	0.67	7.32E-03

Figures 3.8 and 3.9 show a comparison of the measured data and fitted curves using Eq. (3.18) and the parameters shown in Table 3.4. It can be seen that the curves using Eq. (3.18) do not fit the test data as closely as those shown in Figs. 3.5 and 3.7 using Eq. (3.16).

Although Eq. (3.18) describes the physical process of slip-sinkage better than Eq. (3.16), considering the method for applying the slip-sinkage equation, which will be discussed in detail in Chapter 4, and their goodness-of-fit, Eq. (3.16) will be used in the tire-terrain interaction models.

Bevameter Internal Shear Tests in Sand

Figure 3.8 Internal Slip Sinkage-Shear Displacement Data Fitted with Eq. (3.18)

Bevameter Rubber Shear Tests in Sand

Figure 3.9 Rubber Slip Sinkage-Shear Displacement Data Fitted with Eq. (3.18)

Chapter 4

Tire–Terrain Interaction Models

A tire-terrain interaction model takes into account both the tire characteristics, such as tire dimensions and inflation pressure, and the terrain properties, such as pressure-sinkage parameters and shearing characteristics. From the model the contact geometry, normal pressure and shear stress along the tire-terrain interface for a given load and tire slip can be predicted. Based on the normal pressure and shear stress distributions, the tractive performance, such as thrust, motion resistance, drawbar pull and sinkage, can be predicted. In this chapter two tire terrain interaction models are presented. One is based on semi-empirical approach. Its formulation is relatively simple, and yet it predicts the tractive performance reasonably well. This model is, therefore, referred to as the “practical model”. The other is a sophisticated model using the finite element method and gives better prediction than that by the practical model but demands much more computing power.

4.1 Practical Model

In the practical model the tire operating over deformable terrain is either in a rigid mode or an elastic mode depending on the terrain rigidity and tire inflation pressure as illustrated in Fig. 4.1. The major dimensions of the tire, such as diameter, width and lug depth, are taken into account in the model. The complex lug patterns on the tire carcass are simplified to an area ratio, by which the shear stresses due to internal

Figure 4.1 Illustration of Operating Modes for the Practical Tire-Terrain Interaction Model

shearing and rubber-terrain shearing are proportioned.

4.1.1 Prediction of Sinkage and Shear Displacement

The terrain model is based on the semi-empirical relationships derived from the beamometer test data, as described in the previous chapter. The pressure-sinkage relationship and the shear stress-shear displacement relationship are repeated here for clarity.

The pressure-sinkage relationship is as follows,

$$p = \left(\frac{k_c}{b} + k_\phi \right) z^n \quad (4.1)$$

The shear stress-shear displacement relationship is shown below,

$$\tau = (c + p \tan \phi)(1 - e^{-\frac{j}{k}}) \quad (4.2)$$

The values of parameters in Eqs. (4.1) and (4.2) are given in Tables 3.1 and 3.2, respectively.

To model the terrain behaviour more closely, the elastic recovery of terrain during unloading is also taken into account. The pressure during unloading and re-loading is assumed to vary linearly [35] (see Fig. 4.2),

$$p = p_u - K_u(z_u - z) \quad (4.3)$$

where $K_u = A_0 + A_u z_u$, whose value is obtained from experimental data, p_u and z_u are the pressure and sinkage at the beginning of unloading, respectively.

To utilize Eqs. (4.1) to (4.3), the values of sinkage and shear displacement must be calculated first, which depend on the tire geometry. Figure 4.3 illustrates the tire model operating in different modes. For the tire operating in the rigid mode as shown in Fig. 4.3a, sinkage z and shear displacement j for $\theta_1 \leq \theta \leq \theta_0$ are calculated as follows,

$$z = R(\cos \theta - \cos \theta_0) \quad (4.4)$$

$$j = R[\theta_0 - \theta - (1 - i)(\sin \theta_0 - \sin \theta)] \quad (4.5)$$

Figure 4.2 Illustration of Terrain Repetitive Loading Characteristics

(a) Shear Displacement in Rigid Mode

(b) Shear Displacement in Elastic Mode

Figure 4.3 Calculation of Shear Displacement for Practical Tire Model

where z is the sinkage at contact angle θ , R is the rolling radius of the tire, θ_0 is the initial contact angle, i is the slip of the tire, and θ is the contact angle for a given point in the contact area and is defined clockwise positive as shown in Fig. 4.3.

For the tire operating in the elastic mode as shown in Fig. 4.3b, the expressions for calculating sinkage z and shear displacement j take different forms in different contact regions.

For $\theta_2 \leq \theta \leq \theta_0$, the expressions are the same as that for rigid mode,

$$z = R(\cos \theta - \cos \theta_0) \quad (4.6)$$

$$j = R[\theta_0 - \theta - (1 - i)(\sin \theta_0 - \sin \theta)] \quad (4.7)$$

For $\theta_3 < \theta < \theta_2$, i.e., θ is in the deflection portion,

$$z = R(\cos \theta_2 - \cos \theta) \quad (4.8)$$

$$j = j_1 + i(R - \delta)(\tan \theta_2 - \tan \theta) \quad (4.9)$$

where θ_2 is the front contact angle of the deflected portion of the tire, j_1 is the shear displacement at θ_2 calculated from Eq. (4.7) and δ is the deflection of the tire as shown in Fig. 4.3b.

For $\theta_1 \leq \theta \leq \theta_3$, i.e., θ is in the region of recovery,

$$z = R(\cos \theta_3 - \cos \theta) \quad (4.10)$$

$$j = j_2 + R[\theta_3 - \theta - (1 - i)(\sin \theta_3 - \sin \theta)] \quad (4.11)$$

where θ_3 is the rear contact angle of the deflected portion of the tire and j_2 is the shear displacement calculated from Eq. (4.9) at θ_3 .

4.1.2 Prediction of Slip-Sinkage for Tire

The slip-sinkage phenomenon has long been observed but seldom incorporated in any tire-terrain interaction models. An effort has been made for incorporating the slip-sinkage into the tire-terrain interaction models in this thesis. It was found during the course of study that whether or not the slip-sinkage is taken into account affects the prediction of tractive performance significantly. In Chapter 3 a relationship between slip-sinkage and shear displacement was derived based on the bevameter shear test

data. The application of this relationship to the analysis of tire-terrain interaction will be explored.

In studying the tractive performance of a tire operating over deformable terrain in steady-state, the equilibrium conditions are used to determine the sinkage, motion resistance, torque, thrust, drawbar pull, etc. Usually, the slip and tire load are input parameters. The horizontal and moment equilibrium conditions are used for determining drawbar pull and torque. The vertical equilibrium condition can be used to determine the total sinkage at the wheel center. The total sinkage of a tire can be considered as composed of a sinkage due to normal load and slip-sinkage due to shearing as described previously.

Figure 4.4 depicts a tire rolling from one steady state through a transient state towards another steady state, when slip-sinkage is taken into account. In the transient state, the motion of the tire can be described using three velocities, the horizontal, vertical and angular velocities. The horizontal velocity is related with the angular velocity through tire radius and slip, and the angular velocity is not related to the sinkage and shear displacement. However, it is necessary to know the vertical velocity profile from state I to state II in order to determine the slip-sinkage.

If the process noted above (two-equilibrium-position approach) is accepted, then the vertical or slip-sinkage velocity should start with zero and end with zero. A sine function is one of the simplest functions that satisfy the conditions. With a sine function, the vertical velocity profile can be described as follows,

$$\frac{dz_{so}}{dt} = A_c \sin(B_c \theta) \frac{d\theta}{dt} \quad (4.12)$$

where A_c and B_c are constants to be determined from the boundary conditions.

For convenience, assuming that the slip-sinkage process completes after the tire rotates an angle of θ_0 , and the wheel center gains a slip-sinkage of Z_0 , Eq. (4.12) then becomes,

$$V_z = \frac{dz_{so}}{dt} = \frac{Z_0 \pi}{2\theta_0} \sin\left(\frac{\pi}{\theta_0} \theta\right) \frac{d\theta}{dt} \quad (4.13)$$

It can be seen from Eq. (4.13) that when $\theta = 0$ or $\theta = \theta_0$, the slip-sinkage velocities are zero. This means the initial and final conditions of the slip-sinkage process are

$$\longleftrightarrow V = R(1-i)\omega$$

Equilibrium II or State II

Transient state

Equilibrium I or State I

Figure 4.4 Illustration of Different States for Two-Equilibrium-Position Approach

satisfied. During the course of study, many forms of the function for describing the vertical velocity profile were studied. It was found that the forms of the function affect the calculation of slip-sinkage along the tire-terrain interface. In this thesis, only one form of the function is presented. It is desirable to search for the exact form of the function if the two-equilibrium-position approach is to be used.

It is noticed in Fig. 4.4 that the contact area increases as the tire sinks deeper. This implies that the pressure and shear stress distributions that maintained the initial equilibrium must be altered during the slip-sinkage process to achieve the final equilibrium.

One way of redistributing the normal pressure distribution is using the slip-sinkage equation proposed, Eq. (3.16) or Eq. (3.18). Taking Eq. (3.16) as an example, the normal pressure can be readily computed if the slip-sinkage z_s and shear displacement j at a given point in the contact area are known,

$$p = \frac{1}{B_s} \left(\frac{z_s}{1 - e^{-\frac{j}{K_s}}} - A_s \right) \quad (4.14)$$

Since the vertical motion of the tire is independent of the horizontal travel and rotation of the tire, the slip-sinkage of any given point on the tire is identical to the slip-sinkage at the wheel center, z_{so} . Integration of Eq. (4.13) with respect to time yields the following function,

$$z_{so} = \frac{Z_0}{2} \left[1 - \cos\left(\frac{\pi}{\theta_0} \theta\right) \right] \quad (4.15)$$

The shear displacement j can be computed using Eq. (4.5) during the process.

If only the final equilibrium state, i.e., after the slip-sinkage has fully developed, is considered, all the slip-sinkages in the contact area are then equal to Z_0 by setting $\theta = \theta_0$ in Eq. (4.15). It is not difficult to find that by substituting Z_0 into Eq. (4.14) the normal pressure is infinite when j equals zero, and decreases as j increases, and approaches a constant value when j is very large. This type of normal pressure distribution is unrealistic. Consequently, the tractive performance prediction based on such a normal pressure distribution will be unreasonable.

To improve the prediction of normal pressure distribution, the transient motion of the tire should be considered. After the tire has rotated an angle of θ , the slip-sinkage

and the shear displacement at contact angle θ are calculated using the following expressions,

$$z_s = \frac{Z_0}{2} \left[1 + \cos\left(\frac{\pi}{\theta_0} \theta\right) \right] \quad (4.16)$$

$$j = R[\theta_p - \theta - (1 - i)(\sin \theta_p - \sin \theta)] \quad (4.17)$$

where θ_p is the contact angle at the instant when the slip-sinkage at wheel center is z_{so} , which is determined from,

$$\theta_p = \cos^{-1} \left(1 - \frac{Z + z_{so}}{R} \right) \quad (4.18)$$

Then the normal pressure at contact angle θ is calculated using the newly obtained values of z_s and j , and this normal pressure will be carried over to the final equilibrium position, where the total reaction and tractive performance can be evaluated.

It was found that this method gives a better normal pressure distribution than the previous one. However, it was also found that this approach is only suitable for two-dimensional analysis of tire-terrain interaction. For three-dimensional analysis, it is required to predict the normal pressure on the sidewall of the tire. If this approach is followed, the the normal pressure on the sidewall will be larger, sometimes much larger, than the normal pressure on the tread. This is because the slip-sinkage only depends on the slip-sinkage velocity at the wheel center, any change in radius does not affect its value. On the other hand, any change in radius affects the value of shear displacement significantly. As a point on the sidewall in the contact region has a smaller radius than that on the tread, the pressure on the sidewall calculated from Eq. (4.14) is greater than that on the tread. This does not appear to be reasonable.

The above method for determining the slip-sinkage in the contact region is based on the analysis of the kinematics of the tire. It does not take into account the physical aspects of slip-sinkage. From the bevameter shear tests and those conducted by other researchers [28, 29], it was observed that the direction of slip-sinkage is perpendicular to its shear displacement. If this observation is incorporated into the above analysis, the vertical slip-sinkage velocity at a point in the contact region can be resolved into normal and tangential components, as shown in Fig. 4.5. It is clear that this type of treatment is only useful for transient analysis.

Figure 4.5 Method for Calculating Slip-Sinkage Using Normal Component

The slip-sinkage at contact angle θ is then the integration of the normal component of the slip-sinkage velocity, which is shown below,

$$\begin{aligned} z_s &= \int_0^{t_p} V_z \cos \theta dt \\ &= \int_\theta^{\theta_p} \frac{Z_0 \pi}{2\theta_0} \sin\left(\frac{\pi}{\theta_0} \theta\right) \cos \theta d\theta \end{aligned} \quad (4.19)$$

where t_p is the time corresponding to the initial contact angle θ_p , when the tire rotation angle is θ .

The corresponding shear displacement is,

$$\begin{aligned} j &= \int_0^{t_p} V_z \sin \theta dt + \int_0^{t_p} V_t dt \\ &= \int_\theta^{\theta_p} \frac{Z_0 \pi}{2\theta_0} \sin\left(\frac{\pi}{\theta_0} \theta\right) \sin \theta d\theta + \int_\theta^{\theta_p} R[1 - (1 - i) \cos \theta] d\theta \end{aligned} \quad (4.20)$$

The normal pressure distribution calculated using this group of z_s and j is similar to the previous one, but the magnitudes of the pressures are less than those from the previous method because z_s is the integration of a component of z_{so} . This means that for a given Z , the current method requires larger Z_0 to establish the final equilibrium than the previous one. Numerical simulations using the current method show that because of this, the contact area at the final equilibrium state is overestimated, and consequently the thrust and drawbar pull are overestimated as well.

For three-dimensional analysis, the above method still has the similar problem to the previous one, i.e., the pressure on sidewall is greater than that on the tread, although it is not as serious as the previous one.

So far the analysis of slip-sinkage has been focused on the two-equilibrium-position approach. Methods derived from this analysis for calculating the normal pressure and shear stress distributions seem to be logical and yield reasonably good predictions of tractive performance of a tire for two-dimensional cases.

The other approach for determining slip-sinkage assumes that there exists a relationship between the sinkage and slip-sinkage. In other words, using this relationship and the vertical equilibrium condition, the sinkage and slip-sinkage at the wheel center can be determined, and the tractive performance can be evaluated.

Again, from slip-sinkage tests it was observed that the slip-sinkage process can be considered as a quasi-static process, i.e., during the slip-sinkage process the normal

pressure and shear stress are nearly unchanged. To demonstrate the concepts that are involved in this approach, it is necessary to start with a simple case, in which a rigid plate with non-uniform loading is studied. Some useful concepts can be developed.

Figure 4.6b shows a plate under non-uniformly distributed load and towed in the horizontal direction. Under such conditions there is no doubt that the plate will sink and change its attitude. The slip-sinkage at a point under the plate may be predicted using Eq. (4.14) if the applied pressure at that point is used. This type of slip-sinkage is named "predicted slip-sinkage". It is not surprising that the predicted slip-sinkages along the plate under different pressures may not vary linearly. However, it is obvious that for a rigid plate the slip-sinkage can only be in a linear fashion, as the plate is assumed to be rigid. To distinguish it from the "predicted slip sinkage", this type of slip-sinkage is termed "geometrically compatible slip-sinkage". Since there exist two types of slip-sinkages, a question could be raised as to how the plate under such a pressure distribution would actually sink, and whether there exists a relationship between the predicted slip-sinkage and geometrically compatible slip-sinkage.

In general, predicted slip-sinkage is a known quantity if the normal pressure and shear displacement are given. On the other hand, geometrically compatible slip sinkage can be considered as a unknown quantity. In fact the distribution of the geometrically compatible slip-sinkage is determined by the geometry, only the magnitude is unknown. Therefore, it is possible to determine the magnitude of the geometrically compatible slip-sinkage using the predicted slip-sinkage. One way of determining the magnitude of geometrically compatible slip-sinkage is by assuming that the average of geometrically compatible slip-sinkage is equal to the average of the predicted slip sinkage. In the case of a rigid plate, the magnitude is the angle of inclination of the plate. The method can be further extended to take into account the influence of pressure on slip-sinkage. This leads to the concept of work equivalence, which is the total work done by the distributed load along the predicted slip sinkage is equal to the total work done by the same load along the geometrically compatible slip sinkage. It is now clear that with this relationship coupled with equilibrium conditions, the tire-terrain interaction problem becomes solvable.

The methods for calculating geometrically compatible slip-sinkage are illustrated in Figs. 4.6c and 4.6d. To define the slip-sinkages, the position of the plate before

Figure 4.6 Illustration of Slip-Sinkage for a Rigid Plate

slip-sinkage is brought forward so that the first contact points coincide and their positions can be compared. One of the methods defines the geometrically compatible slip-sinkage of a point on the plate as the displacement in the normal direction, which is $z_s = s\theta$. The other defines it as the vertical displacement measured from the position after slip-sinkage to the position before slip-sinkage, which is $z_s = s \cos \theta$.

To analyze the slip-sinkage for tire, the tire position before slip-sinkage (shown in dotted line) is brought forward so that the first contact points coincide at point *A*, as shown in Figs. 4.7a and 4.7b. Following the concept developed for a rigid plate, the slip-sinkage, i.e., geometrically compatible slip-sinkage, at a point in the contact area of a tire can be calculated.

The method for calculating the slip-sinkage in the normal direction is shown in Fig. 4.7a. In this method the contact area remains unchanged during the slip-sinkage process. Points in the contact area before slip-sinkage AC_1 correspond to points in the contact area after slip-sinkage AC_2 . The terrain surface S_1 due to sinkage Z is further compacted to surface S_2 due to slip-sinkage Z_0 , so that at equilibrium the total sinkage at the wheel center O' is $Z + Z_0$. It should be noted, however, that since at sinkage Z the tire is not necessarily in equilibrium, surface S_1 is not an elastic recovery surface. A point P on AC_1 will be at P' after the wheel center has gained a slip-sinkage of Z_0 . It can be shown that the normal displacement of P is the distance between point D and point P' . This normal displacement of P can be defined as the slip-sinkage of point P , which is similar to the slip-sinkage in the normal direction for a plate as discussed earlier.

A close examination of the above definition reveals that it limits the maximum possible slip-sinkage at the wheel center for a given sinkage and tire configuration. This can be seen from Fig. 4.8a, as the slip-sinkage at the wheel center Z_0 increases, the terrain surface before slip-sinkage S_1 must move up to maintain the contact area AC_1 the same as AC_2 . However, the highest position point C_1 can go is point B , which is at the same level as point A . Any positions of C_1 above B are unacceptable. This is because the normal pressure, which is calculated at sinkage Z and assumed to be unchanged during the slip-sinkage process, in the region near point C_2 can not be obtained if C_1 is higher than B . Thus, there exists a limit for the maximum possible slip-sinkage.

Figure 4.8 Illustration of Limits on Methods for Calculating Slip-Sinkages

Figure 4.8a shows how the limit can be determined. The angle 2ϕ represents the largest possible contact angle after slip-sinkage for a given Z . It can be seen that if Z increases or decreases, the angle ϕ and length AC_1 (or AB , or AC_2) will increase or decrease accordingly. This means the slip-sinkage at the wheel center Z_0 is restricted by the given sinkage Z . The mathematical expression for such a limit is as follows,

$$\begin{aligned}
 Z_0 &= \overline{OD} - (R - \overline{AC_1} \sin \phi) \\
 &= \overline{OD} - (R - \overline{AB} \sin \phi) \\
 &= (R - Z) - [R - (2R \sin \phi) \sin \phi] \\
 &= 2R \sin^2 \phi - Z \\
 &= 2R \left[\frac{Z(2R - Z)}{R^2} \right] - Z \\
 &= \frac{Z(3R - 2Z)}{R}
 \end{aligned} \tag{4.21}$$

where R is the radius of the tire.

The above equation states correctly that when the sinkage Z is zero, the slip-sinkage at the wheel center Z_0 is also zero. But when the sinkage is $\frac{R}{4}$, the maximum possible slip-sinkage is only $\frac{5R}{8}$. This kind of limitation to slip-sinkage imposed purely by geometry, and not by the terrain and tire loading conditions, does not appear to be reasonable. This is because it is possible, under certain conditions, that the slip-sinkage at the wheel center, as determined from the work equivalence assumption, may exceed the maximum possible slip-sinkage. It is clear that modification to the definition of slip-sinkage at a point in contact area of a tire is necessary.

Figure 4.7b shows the modified definition of slip-sinkage at a point in the contact area. It is defined as the distance between P and P' , where points P and P' are on the line passing through center O . This slip-sinkage is termed as the slip-sinkage in the radial direction, which is similar to the slip-sinkage in the vertical direction for a plate as discussed earlier. Unlike the first definition the point P' is not the same physical point of P after slip-sinkage. The calculation of slip-sinkage is approximate in comparison with the previous one. However, this definition relaxes the limitation to the slip-sinkage at the wheel center considerably. Figure 4.8b illustrates such a relaxation. Considering triangles $\overline{AOC}$ and $\overline{C'O'C_2}$, and using sine law, the following

relations can be obtained,

$$\frac{R}{\sin \beta} = \frac{\overline{CO'}}{\sin \phi} = \frac{R - \overline{CO'}}{\sin 2\phi} \quad (4.22)$$

From Eq. (4.22), one can obtain,

$$Z_0 = R[\cos \phi + \cos(\beta + \phi)] \quad (4.23)$$

where $\beta = \sin^{-1}[(2 \cos \phi + 1) \sin \phi]$, $\frac{\pi}{2} \leq \beta \leq \pi$.

Equation (4.23) is valid only when $(2 \cos \phi + 1) \sin \phi \leq 1$, which requires the value of angle ϕ must not be greater than 20.35° . The corresponding sinkage Z for this angle is $0.0624R$ (approximately 6% of the tire radius), and the slip-sinkage Z_0 is $0.5898R$. The ratio of Z_0 to Z is 9.4, which is much larger than the ratio of 2.5 based on the previous definition. It can be shown that the ratio decreases as the value of ϕ decreases. When ϕ is zero, the ratio is 3. For any value of ϕ greater than 20.35° , Eq. (4.23) will no longer be valid. In other words, for a given sinkage Z , if ϕ is greater than 20.35° , then there is no limitation to Z_0 in the sense of practical application.

From the above analysis of the definition of slip-sinkage, it is clear that the latter definition of slip-sinkage should be used for predicting the slip-sinkage at the wheel center.

Figure 4.9 shows how the slip-sinkage at a point in the contact region of a tire in the rigid mode is calculated,

$$\begin{aligned} z_s &= \overline{OP'} - R \\ &= \frac{R \cos \gamma + Z_0}{\cos \theta} - R \end{aligned} \quad (4.24)$$

where R is the radius of the tire, θ and γ are the contact angles of point P before and after slip-sinkage occurs, respectively, and Z_0 is the slip-sinkage at the wheel center.

The relations between θ and γ are,

$$\theta = \tan^{-1} \left(\frac{R \sin \gamma - X_L}{R \cos \gamma + Z_0} \right) \quad (4.25)$$

$$\gamma = \sin^{-1} \left(\frac{Z_0 \sin \theta + X_L \cos \theta}{R} \right) + \theta \quad (4.26)$$

where X_L is the horizontal distance between center O and center O' .

Figure 4.9 Definition of Slip-Sinkage for Rigid Mode

The X_L is computed as follows,

$$X_L = R(\sin \gamma_0 - \sin \theta_0) \quad (4.27)$$

where γ_0 and θ_0 are the initial contact angles as shown in Fig. 4.9, and can be calculated from the geometry,

$$\theta_0 = \cos^{-1}\left(1 - \frac{Z}{R}\right) \quad (4.28)$$

$$\gamma_0 = \cos^{-1}\left(1 - \frac{Z + Z_0}{R}\right) \quad (4.29)$$

From Eqs. (4.24) to (4.29), it can be seen that the geometrically compatible slip-sinkage at a given point P in the contact area is uniquely defined by geometric parameters Z , Z_0 and R .

At this point, it may be necessary to summarize the methods described above. In the view of author, both approaches, whether two-equilibrium-position with assumed path function or one-equilibrium-position with assumed work equivalence relationship, have certain discrepancies. In other words, some additional assumptions have to be made in order to proceed with the analysis and to eventually incorporate the slip-sinkage effect into the tire-terrain interaction model.

For the two-equilibrium-position approach, in addition to the final equilibrium, an equilibrium before slip-sinkage has to be assumed to exist. Furthermore, a path function between the initial equilibrium before slip-sinkage and the final after slip-sinkage has to be assumed. Since physically there may not exist two equilibrium positions for a single process, such path function is difficult to verify experimentally. Thus, the path function selected can only satisfy initial and final conditions. Moreover, when applying this approach in modeling tire-terrain interaction in three dimensions, it gives unrealistic predictions of normal pressure on sidewalls. Due to these drawbacks this approach is not used in the tire-terrain interaction models. It is presented in this thesis for comparing with the other approach, and for providing ideas for other researchers who may be interested in this problem.

For the one-equilibrium-position approach, a relationship between the sinkage and slip-sinkage at the wheel center is assumed to exist, and it is the work equivalence relationship. This relationship is also difficult to verify experimentally, since it is

impossible to separate sinkage or slip-sinkage from total sinkage. Because the slip-sinkage for a point in the contact area is obtained by comparing the changes in position of a given point on the tire, it can be termed as apparent slip-sinkage. Since the slip-sinkage process is assumed to be quasi-static, this approach will not give any unrealistic predictions of normal pressure. The normal pressure and shear stress calculated at sinkage Z are maintained. Comparing to the two-equilibrium-position approach, this approach seems better, and it is used in the tire-terrain interaction models adopted.

Since certain assumptions are required to calculate the slip-sinkage, it can not be certain which of the proposed methods is the closest representation of the true slip-sinkage. However, studies show that slip-sinkages calculated using these methods are not far from each other. This is illustrated in Figs. 4.10, 4.11 and 4.12, where for demonstration purposes the sinkage at the wheel center is set at 5 cm and the slip-sinkage at the wheel center are selected at 1 cm, 5 cm and 12 cm, respectively. It can be seen from these figures that when the slip-sinkage at the wheel center is small relative to its sinkage, three slip-sinkage curves are very close to one another. As the slip-sinkage at the wheel center increases, these three curves move away from each other. However, the apparent slip-sinkage in the radial direction is always between the slip-sinkage obtained from the integration of slip-sinkage velocity and the apparent slip-sinkage in the normal direction.

Thus far, the analysis of slip-sinkage has been focused on tire operating in the rigid mode, that is, the tire is assumed to be perfectly round. Under certain conditions, such as at low tire inflation pressures, high applied load and soft terrain condition (see Fig. 4.1), part of tire will deflect. Previous studies on tire-terrain interaction assume that the deflected portion of the lowest part of the tire is flat and level if the vehicle is operating over level terrain, as illustrated in Fig. 4.1 and Fig. 4.13 (shown in dotted line). This assumption implies the maximum normal pressure is at the lowest point and vertical. However, experimental data show that the maximum normal pressure is usually not at the lowest point, rather it is in front of the lowest point, and its direction is not necessarily vertical [6, 7, 41]. It is reasonable to speculate that the position of the maximum deflection of the tire should be close to the maximum normal pressure. Therefore, the deflected portion of the tire should follow the maximum

Michelin 11.00R16 XL On Sand

Figure 4.10 Comparisons of Slip-Sinkages
for $Z = 5$ cm and $Z_0 = 1$ cm

Michelin 11.00R16 XL On Sand

Figure 4.11 Comparisons of Slip-Sinkages
for $Z = 5$ cm and $Z_0 = 5$ cm

Michelin 11.00R16 XL On Sand

Figure 4.12 Comparisons of Slip-Sinkages
for $Z = 5$ cm and $Z_0 = 12$ cm

normal pressure and be perpendicular to the maximum normal pressure.

Based on such speculation, the deflected tire after slip-sinkage is assumed to have a geometry and position, as shown in Fig. 4.13 in solid line. Such situation can be obtained by considering the deflected tire at sinkage Z gains a slip-sinkage of Z_0 at the wheel center after one or several complete revolutions, and the initial contact points coincide at point A , and the relative position between point A and the deflected portion remains. In other words, because of the slip-sinkage the tire changes its attitude and has an equivalent rotation about point A such that the slip-sinkage at the wheel center is Z_0 .

The definition of apparent slip-sinkage in the radial direction is adopted for the tire in the elastic mode for the same reason as discussed previously for the tire in the rigid mode. The slip-sinkage for the undeflected portion is then the same as that for rigid mode. For a given point in the deflected region, the slip-sinkage is calculated as follows,

$$z_s = \frac{R \cos \gamma + Z_0 - (R - d)}{\cos \theta} - \frac{R \cos(\gamma - \gamma_3 - \theta_2) - (R - \delta)}{\sin(\frac{\pi}{2} - \theta + \gamma_3 + \theta_2)} \quad (4.30)$$

where θ_2 and γ_3 are the contact angles as shown in Fig. 4.13, and δ is the deflection of the tire.

In the region of $\theta_2 < \theta < \theta_0$ and $\gamma_3 < \gamma < \gamma_2$, the slip-sinkage takes the following form,

$$z_s = \frac{R \cos \gamma + Z_0}{\cos \theta} - R - \frac{R \cos(\gamma - \gamma_3 - \theta_2) - (R - \delta)}{\sin(\frac{\pi}{2} - \theta + \gamma_3 + \theta_2)} \quad (4.31)$$

In the region of $\theta_3 < \theta < \theta_2$ and $\gamma_1 < \gamma < \gamma_3$, the slip-sinkage takes the following form,

$$z_s = \frac{R \cos \gamma + Z_0 - (R - \delta)}{\cos \theta} \quad (4.32)$$

It is noticed that the slip-sinkage for the rigid mode is always perpendicular to the contact surface before slip-sinkage occurs, while the slip-sinkage for the elastic mode is not always perpendicular to the original contact surface. The reason for choosing such a calculation is to maintain the continuity of slip sinkage across the entire contact area.

4.1.3 Criterion for Determining the Operating Mode of a Tire

To determine whether a tire is operating in rigid mode or elastic mode, the concept of nominal pressure is used, which is defined as follows [4],

$$p_n = \eta_l p_l + (1 - \eta_l) p_c \quad (4.33)$$

where p_n is the nominal pressure, p_c is the pressure acting on tire carcass, p_l is the pressure acting on tire lugs and η_l is the ratio of lug area to carcass area.

In conjunction with the nominal pressure, the concept of ground pressure plays an essential role in determining the operating mode of the tire, which is defined as follows,

$$p_g = \frac{N_L}{A_p} \quad (4.34)$$

where p_g is the ground pressure, N_L is the normal load applied when the tire is on level hard ground and A_p is the projected contact area.

The relationship between the projected contact area and the normal load is obtained in the static test. For a given normal load the ground pressure can be uniquely determined from this relationship. Therefore, the method for determining the operating mode of the tire, which is best illustrated in Fig. 1.1, is:

- determine the sinkage at which the nominal pressure on the tire is equal to the ground pressure. The sinkage is called transition sinkage.
- evaluate the vertical reaction from the terrain on the tire at transition sinkage assuming the tire is in the rigid mode.
- if the vertical reaction is greater than the normal load applied on the tire, which means the vertical equilibrium has been established before the transition sinkage is reached, then the operating mode of the tire is rigid.
- if the vertical reaction is less than the normal load, which means the tire must deflect to increase the contact area to establish vertical equilibrium, then the operating mode of the tire is elastic.

Once the tire is operating in the elastic mode, the nominal pressure beneath the deflected portion of the tire remains the same as the ground pressure for a given load.

4.1.4 Prediction of Shear Stress under Continuously Varying Normal Pressure

It was first conceived by Wong and Preston-Thomas [4] that the shear stress-shear displacement relation, Equation (4.2), is of path-dependent nature for frictional shearing. To take into account the effect of the variation of normal pressure on the calculation of shear stress, a special numerical procedure was employed.

Figure 4.14a shows three shear stress-shear displacement curves under three constant normal pressures. As described in Chapter 3, Eq. (4.2) is derived from the shear tests under constant normal loads. In reality, however, the normal pressure in the contact area varies continuously, Eq. (4.2) may not be correct for calculating the shear stress under such conditions. It was shown by Wong and Preston-Thomas [4] that the shear stress under continuously varying normal pressure should follow the path shown in Fig. 4.14b. A special step by step method was used to calculate the shear stress at a given shear displacement. It was found during the course of study that this special method can be expressed in a more concise mathematical form.

To achieve this, the shear stress τ_1 for pressures p_1 and p_2 is expressed as follows,

$$\tau_1 = p_1 \tan \phi (1 - e^{-\frac{j_1}{K}}) \quad (4.35)$$

$$\tau_1 = p_2 \tan \phi (1 - e^{-\frac{j_2}{K}}) \quad (4.36)$$

Re-arranging Eq. (4.36), one obtains,

$$e^{-\frac{j_2}{K}} = 1 - \frac{\tau_1}{p_2 \tan \phi} \quad (4.37)$$

Shear stress τ_2 is expressed as follows,

$$\begin{aligned} \tau_2 &= p_2 \tan \phi (1 - e^{-\frac{j_2 + \Delta j_2}{K}}) \\ &= p_2 \tan \phi (1 - e^{-\frac{j_2}{K}} e^{-\frac{\Delta j_2}{K}}) \end{aligned} \quad (4.38)$$

where $\Delta j_2 = j'_2 - j_2$.

Substituting Eq. (4.37) into Eq. (4.38), one obtains,

$$\tau_2 = p_2 \tan \phi (1 - e^{-\frac{\Delta j_2}{K}}) + \tau_1 e^{-\frac{\Delta j_2}{K}} \quad (4.39)$$

Figure 4.14 Illustration of Re-Shearing Concept for Frictional Terrain

Since Δj_2 is small, $e^{-\frac{\Delta p}{K}}$ can be approximated as,

$$e^{-\frac{\Delta p}{K}} = 1 - \frac{\Delta j_2}{K} \quad (4.40)$$

Substituting Eq. (4.40) into Eq. (4.39), one obtains,

$$\tau_2 = p_2 \tan \phi \left[1 - \left(1 - \frac{\Delta j_2}{K} \right) \right] + \tau_1 \left(1 - \frac{\Delta j_2}{K} \right) \quad (4.41)$$

Re-arranging Eq. (4.41), it becomes,

$$\tau_2 - \tau_1 = p_2 \tan \phi \frac{\Delta j_2}{K} - \tau_1 \frac{\Delta j_2}{K} \quad (4.42)$$

or

$$\frac{\Delta \tau}{\Delta j_2} = \frac{p_2 \tan \phi}{K} - \frac{\tau_1}{K} \quad (4.43)$$

Δj_2 in Eq. (4.43) can be re-written as,

$$\Delta j_2 = \Delta j + \Delta j_1 \quad (4.44)$$

where

$$\Delta j_1 = j_1 - j_2 \quad (4.45)$$

Substituting Eq. (4.44) into Eq. (4.43), one obtains,

$$\frac{\Delta \tau}{\Delta j + \Delta j_1} = \frac{p_2 \tan \phi}{K} - \frac{\tau_1}{K} \quad (4.46)$$

Taking the limit of the left-hand side of Eq. (4.46), one obtains,

$$\begin{aligned} \lim_{\Delta j \rightarrow 0} \frac{\Delta \tau}{\Delta j + \Delta j_1} &= \frac{\lim_{\Delta j \rightarrow 0} \Delta \tau}{\lim_{\Delta j \rightarrow 0} \Delta j + \lim_{\Delta j \rightarrow 0} \Delta j_1} \\ &= \frac{d\tau}{dj + \lim_{\Delta j \rightarrow 0} \Delta j_1} \end{aligned} \quad (4.47)$$

To evaluate the remaining limit in Eq. (4.47), first eliminating τ_1 from Eqs. (4.35) and (4.36) and solving for j_2 , one obtains,

$$j_2 = -K \ln \left[1 - \frac{p_1}{p_2} (1 - e^{-\frac{j_1}{K}}) \right] \quad (4.48)$$

From Eq. (4.48) it can be seen that as Δj approaches zero, p_2 approaches p_1 as well and j_2 approaches j_1 at the same time. In the limit,

$$\begin{aligned}
 \lim_{\Delta j \rightarrow 0} j_2 &= \lim_{\Delta j \rightarrow 0} \left\{ -K \ln \left[1 - \frac{p_1}{p_2} (1 - e^{-\frac{\Delta j}{K}}) \right] \right\} \\
 &= \lim_{p_2 \rightarrow p_1} \left\{ -K \ln \left[1 - \frac{p_1}{p_2} (1 - e^{-\frac{\Delta j}{K}}) \right] \right\} \\
 &= -K \ln \left[1 - \frac{p_1}{p_1} (1 - e^{-\frac{\Delta j}{K}}) \right] \\
 &= -K \ln \left[1 - (1 - e^{-\frac{\Delta j}{K}}) \right] \\
 &= -K \ln (e^{-\frac{\Delta j}{K}}) \\
 &= -K \left(-\frac{j_1}{K} \right) \\
 &= j_1
 \end{aligned} \tag{4.49}$$

Therefore, $\lim_{\Delta j \rightarrow 0} \Delta j_1 = 0$, Eq. (4.47) becomes,

$$\lim_{\Delta j \rightarrow 0} \frac{\Delta \tau}{\Delta j + \Delta j_1} = \frac{d\tau}{dj} \tag{4.50}$$

Taking the limit of the right-hand side of Eq. (4.46), one obtains,

$$\begin{aligned}
 \lim_{\Delta j \rightarrow 0} \left(\frac{p_2 \tan \phi}{K} - \frac{\tau_1}{K} \right) &= \lim_{p_2 \rightarrow p_1} \frac{p_2 \tan \phi}{K} - \lim_{\tau_2 \rightarrow \tau_1} \frac{\tau_1}{K} \\
 &= \frac{p \tan \phi}{K} - \frac{\tau}{K}
 \end{aligned} \tag{4.51}$$

Equating Eq. (4.51) to Eq. (4.50), the differential equation relating pressure and shear stress is then,

$$\frac{d\tau}{dj} + \frac{\tau}{K} = \frac{p \tan \phi}{K} \tag{4.52}$$

Re-writing Eq. (4.52) in an integral form,

$$\tau = \frac{\tan \phi}{K} e^{-\frac{j}{K}} \int^j p(\xi) e^{\frac{\xi}{K}} d\xi \tag{4.53}$$

Equation (4.53) gives the exact expression for calculating frictional shear stress under continuously varying normal pressure. Figure 4.15 illustrates an example how the shear stress calculated by the step-by-step method approaches the analytical value as the number of step increases. It is clear that the results obtained using the analytical method are more accurate and efficient than using the step-by-step method.

Figure 4.15 Comparison of Results Obtained Using Analytical and Numerical Procedures for Determining Re-Shearing Effects

4.1.5 Work Equivalence Assumption and Search Scheme

Based on the analysis for slip-sinkage in Section 4.1.2, the work equivalence assumption is applied in modelling tire slip-sinkage. The assumption states that the total work done by the distributed load along the predicted slip-sinkage is equal to the total work done by the same load along the geometrically compatible slip-sinkage. Mathematically it means,

$$\int_A p(z_s - z_s^*)dA = 0 \quad (4.54)$$

where p is the normal pressure, z_s is the geometrically compatible slip-sinkage, z_s^* is the predicted slip-sinkage and A is the contact area.

Equation (4.54) indicates that although there may be two ways for a tire to have slip-sinkage, neither should be different from the other in terms of total energy. In other words, if all the energy is consumed for compacting the terrain, then there should be no difference how the terrain is compacted, whether along the predicted slip-sinkage or geometrically compatible slip-sinkage.

From Eq. (4.54) the slip-sinkage at the wheel center Z_0 can be determined for a given sinkage Z using the numerical technique described. Then the total sinkage at the wheel center can be determined from the vertical equilibrium condition of the tire, i.e.,

$$W = \int R b (\tau \sin \gamma + p \cos \gamma) d\gamma \quad (4.55)$$

where W is the normal load on the tire and b is the contact width of the tire.

Figure 4.16 shows the flowchart for obtaining the vertical equilibrium. The procedure can be summarized as follows:

1. reading the input data, including tire dimensions, terrain parameters, and tire slip.
2. determining the transition sinkage, checking whether the tire is in the rigid mode or elastic mode.
3. if tire is operating in the rigid mode, setting trial value of sinkage at the wheel center, Z . Otherwise, setting trial value of tire deflection, δ .

Figure 4.16 Flowchart of Practical Tire-Terrain Interaction Model

4. calculating the normal pressure and slip-sinkage distributions in the contact region.
5. determining the slip-sinkage at the wheel center, Z_0 , using the method of work equivalence.
6. Evaluating the vertical reaction from the normal pressure and shear stress distributions.
7. checking convergence. If the vertical reaction is balanced by the applied load, the tractive performance of the tire is calculated and printed out, and the program is terminated. If not, the process is repeated.

The tractive performance of the tire can then be evaluated using the following expressions,

$$T = \int R b \tau \cos \gamma d\gamma \quad (4.56)$$

$$R_m = \int R b p \sin \gamma d\gamma \quad (4.57)$$

$$DB = T - R_m \quad (4.58)$$

where T is the thrust generated by the tire, R_m is the motion resistance of the tire, and DB is the drawbar pull.

It is noticed from the above search procedure that if the influence of slip-sinkage is not significant, i.e., the values of parameters A_s and B_s in Table 3.3 is very small, the value of Z_0 will be also small and equilibrium will naturally establish around sinkage Z . In the limit, when the values of A_s and B_s are zero, equilibrium will be at Z . This is another merit of this one-equilibrium-position approach. In the two-equilibrium-position approach the re-calculated normal pressure distribution will not return to the original pressure distribution even if the slip-sinkage is negligible. In such a case the search for slip-sinkage Z_0 is terminated artificially to maintain the initial equilibrium at Z .

4.2 Finite Element Model

This section describes the formulation of a three-dimensional finite element model for a tire in contact with deformable terrain, the behaviour of which is described by a set of semi-empirical equations.

4.2.1 Conversion of the Normal Pressure to Nodal Forces

As is well known, in finite element formulation distributed forces acting on the body in question must be converted to nodal forces before solving the simultaneous linear equations [30].

In the case of study of tire-terrain interaction, there are two kinds of distributed forces acting on the tire. They are normal pressure and shear stress. Methods for converting the pressure and shear stress using the semi-empirical equations described in Chapter 3 are demonstrated here.

Figure 4.17a shows normal pressure acting on a small area $d\vec{A}$ on a spatial surface. The force acting on $d\vec{A}$ is then,

$$d\vec{F} = p d\vec{A} \quad (4.59)$$

The differential area $d\vec{A}$ in Eq. (4.59) is defined as,

$$d\vec{A} = [a_x \vec{i} + a_y \vec{j} + a_z \vec{k}] d\xi d\eta \quad (4.60)$$

where

$$a_x = \frac{\partial y}{\partial \xi} \frac{\partial z}{\partial \eta} - \frac{\partial z}{\partial \xi} \frac{\partial y}{\partial \eta} \quad (4.61)$$

$$a_y = \frac{\partial z}{\partial \xi} \frac{\partial x}{\partial \eta} - \frac{\partial x}{\partial \xi} \frac{\partial z}{\partial \eta} \quad (4.62)$$

$$a_z = \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi} \frac{\partial x}{\partial \eta} \quad (4.63)$$

$\vec{i}$, $\vec{j}$ and $\vec{k}$ are unit vectors in the global coordinate system, x, y and z are global coordinates, and ξ and η are local coordinates defined on the surface.

The components of the total force on the surface are then,

$$F_x = \int_A p dA_x = \int_A p a_x d\xi d\eta \quad (4.64)$$

Figure 4.17 Conversion of Normal Pressure and Shear Stress to Nodal Forces

$$F_y = \int_A p d\Lambda_y = \int_A p a_y d\xi d\eta \quad (4.65)$$

$$F_z = \int_A p d\Lambda_z = \int_A p a_z d\xi d\eta \quad (4.66)$$

To convert the total forces on an element to nodal forces, the shape functions for interpolating the coordinates and the displacements based on the values at nodes are required. In this study, a type of isoparametric element is selected. With isoparametric element, both geometry and normal pressure can be interpolated using the same set of shape functions, N_i , as follows,

$$p(\xi, \eta) = \sum_{i=1}^n N_i(\xi, \eta) p_i \quad (4.67)$$

$$x(\xi, \eta) = \sum_{i=1}^n N_i(\xi, \eta) x_i \quad (4.68)$$

$$y(\xi, \eta) = \sum_{i=1}^n N_i(\xi, \eta) y_i \quad (4.69)$$

$$z(\xi, \eta) = \sum_{i=1}^n N_i(\xi, \eta) z_i \quad (4.70)$$

where n is the number of nodes in an element, p_i , x_i , y_i and z_i are the normal pressure and global coordinates at each node, respectively.

Substituting Eq. (4.67) into Eqs. (4.64) to (4.66), the following expressions can be obtained,

$$F_x = \int_{-1}^{+1} \int_{-1}^{+1} \left[\sum_{i=1}^n N_i(\xi, \eta) p_i \right] a_x d\xi d\eta \quad (4.71)$$

$$F_y = \int_{-1}^{+1} \int_{-1}^{+1} \left[\sum_{i=1}^n N_i(\xi, \eta) p_i \right] a_y d\xi d\eta \quad (4.72)$$

$$F_z = \int_{-1}^{+1} \int_{-1}^{+1} \left[\sum_{i=1}^n N_i(\xi, \eta) p_i \right] a_z d\xi d\eta \quad (4.73)$$

or

$$F_x = \sum_{i=1}^n \int_{-1}^{+1} \int_{-1}^{+1} N_i(\xi, \eta) p_i a_x d\xi d\eta \quad (4.74)$$

$$F_y = \sum_{i=1}^n \int_{-1}^{+1} \int_{-1}^{+1} N_i(\xi, \eta) p_i a_y d\xi d\eta \quad (4.75)$$

$$F_z = \sum_{i=1}^n \int_{-1}^{+1} \int_{-1}^{+1} N_i(\xi, \eta) p_i a_z d\xi d\eta \quad (4.76)$$

In Eqs. (4.74) to (4.76) each double integral represents the force on a node in a particular direction. These integrals are evaluated using Gauss quadrature. Taking the force at the k th ($1 \leq k \leq n$) node in the element in the x -direction, $(F_x)_k$, for example,

$$(F_x)_k = \sum_{i=1}^m \sum_{j=1}^m N_k(\xi_i, \eta_j) p_k a_x w_i w_j \quad (4.77)$$

By substituting Eq. (4.61) into Eq. (4.77), it becomes,

$$(F_x)_k = \sum_{i=1}^m \sum_{j=1}^m N_k(\xi_i, \eta_j) p_k \left(\frac{\partial y}{\partial \xi} \frac{\partial z}{\partial \eta} - \frac{\partial z}{\partial \xi} \frac{\partial y}{\partial \eta} \right) w_i w_j \quad (4.78)$$

where m is the number of Gauss points, w_i and w_j are the weights of Gauss quadrature, and

$$\frac{\partial y}{\partial \xi} = \sum_{i=1}^n \frac{\partial N_i(\xi, \eta)}{\partial \xi} y_i \quad (4.79)$$

$$\frac{\partial z}{\partial \eta} = \sum_{i=1}^n \frac{\partial N_i(\xi, \eta)}{\partial \eta} z_i \quad (4.80)$$

$$\frac{\partial y}{\partial \eta} = \sum_{i=1}^n \frac{\partial N_i(\xi, \eta)}{\partial \eta} y_i \quad (4.81)$$

$$\frac{\partial z}{\partial \xi} = \sum_{i=1}^n \frac{\partial N_i(\xi, \eta)}{\partial \xi} z_i \quad (4.82)$$

The above procedure demonstrates how the applied normal pressure on a spatial surface (an element) can be converted to nodal forces, $(F_x)_k$, $(F_y)_k$ and $(F_z)_k$.

4.2.2 Conversion of the Shear Stress to Nodal Forces

The shear stress shown in Fig. 4.17b can be converted to nodal forces in a similar way to that for pressure. The shear stress over the surface is evaluated using nodal values as follows,

$$\tau(\xi, \eta) = \sum_{i=1}^n N_i(\xi, \eta) \tau_i \quad (4.83)$$

where N_i and τ_i are the shape function and shear stress at node i , respectively.

Using the same area vector as defined by Eq. (4.60), the force acting on $d\vec{A}$ due to shear stress is,

$$d\vec{F} = \vec{\tau} \tau |d\vec{A}| \quad (4.84)$$

The direction vector of the shear stress, $\vec{t}$, and the norm of the vector, $d\vec{A}$, are defined as follows,

$$\vec{t} = [\cos \varphi_x, \cos \varphi_y, \cos \varphi_z]^T \quad (4.85)$$

$$|d\vec{A}| = \sqrt{a_x^2 + a_y^2 + a_z^2} d\xi d\eta = |a| d\xi d\eta \quad (4.86)$$

Following a similar procedure for deriving Eqs. (4.74) to (4.76), the forces on the nodes due to shear stress are,

$$(F_x)_i = \int_{-1}^{+1} \int_{-1}^{+1} N_i(\xi, \eta) \tau_i \cos \varphi_x |a| d\xi d\eta \quad i = 1, 2, \dots, n \quad (4.87)$$

$$(F_y)_i = \int_{-1}^{+1} \int_{-1}^{+1} N_i(\xi, \eta) \tau_i \cos \varphi_y |a| d\xi d\eta \quad i = 1, 2, \dots, n \quad (4.88)$$

$$(F_z)_i = \int_{-1}^{+1} \int_{-1}^{+1} N_i(\xi, \eta) \tau_i \cos \varphi_z |a| d\xi d\eta \quad i = 1, 2, \dots, n \quad (4.89)$$

where n is the number of nodes in the element.

The integrals above are evaluated numerically using Gauss quadrature. Taking the force due to shear stress at k th ($1 \leq k \leq n$) node in the y -direction, $(F_y)_k$, for example,

$$(F_y)_k = \sum_{i=1}^m \sum_{j=1}^m N_k(\xi_i, \eta_j) \tau_k \cos \varphi_y |a| w_i w_j \quad (4.90)$$

In the formulation above only the direction vector remains to be determined, which is dependent on the coordinate system. In this thesis the coordinate system for the tire-terrain interaction model is chosen such that the origin of coordinate system is located at the wheel center, the x axis is parallel to the level ground and in the direction of the translation of the tire, the y axis is vertical, and the z axis is determined using the right-hand rule (see Fig. 2.1).

In such a coordinate system it is apparent that the shear stress must act on the planes parallel to the x - y plane, i.e., in the plane of motion. Therefore, the direction vector of shear stress, $\vec{t}$, becomes,

$$\vec{t} = [\cos \alpha_e, \cos(\frac{\pi}{2} - \alpha_e), 0]^T \quad (4.91)$$

or

$$\vec{t} = [\cos \alpha_e, \sin \alpha_e, 0]^T \quad (4.92)$$

The tangent of angle α_e by definition is,

$$\tan \alpha_e = \frac{\partial y}{\partial x} \quad (4.93)$$

To obtain α_e , the partial derivatives of y with respect to ξ and η , respectively, are taken,

$$\frac{\partial y}{\partial \xi} = \frac{\partial y}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial y}{\partial z} \frac{\partial z}{\partial \xi} \quad (4.94)$$

$$\frac{\partial y}{\partial \eta} = \frac{\partial y}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial z} \frac{\partial z}{\partial \eta} \quad (4.95)$$

Solving $\frac{\partial y}{\partial x}$ from Eqs. (4.94) and (4.95), one obtains,

$$\begin{aligned} \frac{\partial y}{\partial x} &= -\frac{\frac{\partial z}{\partial \eta} \frac{\partial y}{\partial \xi} - \frac{\partial z}{\partial \xi} \frac{\partial y}{\partial \eta}}{\frac{\partial z}{\partial \xi} \frac{\partial x}{\partial \eta} - \frac{\partial z}{\partial \eta} \frac{\partial x}{\partial \xi}} \\ &= -\frac{a_x}{a_y} \end{aligned} \quad (4.96)$$

Therefore, the angle α_e is,

$$\alpha_e = \tan^{-1} \left(-\frac{a_x}{a_y} \right) \quad (4.97)$$

The above procedure demonstrates how the applied shear stress on a spatial surface (an element) can be converted to nodal forces $(F_x)_k$, $(F_y)_k$ and $(F_z)_k$.

4.2.3 Calculation of Shear Displacement

In order to calculate shear stress and slip-sinkage the shear displacement must be known first. The method for calculating shear displacement for the finite element model is described below.

The shear displacement for a deformable tire model as shown in Fig. 4.18a is,

$$j = \int_0^{\theta_0} \left[r(\theta) \cos(\theta - \alpha_e) - R(1 - i) \cos \alpha_e \right] d\theta \quad (4.98)$$

where $\bar{R}$ is defined as the length of the largest circumference of a deformed tire divided by 2π . This $\bar{R}$ may be called "non-slip rolling radius".

(a) Shear Displacement

(b) Slip-sinkage

Figure 4.18 Definitions of Shear Displacement and Slip-Sinkage for FEM Tire Model

Due to the deformation of the tire the integral in Eq. (4.98) can not be evaluated analytically. Therefore, numerical integration has to be used. To perform numerical integration a number of points along the path of shear displacement must be determined.

Although a point on tire surface is described by three coordinates, only two of them are independent. This is because the point is confined to the tire surface, a relationship exists among the three coordinates. Therefore, to locate the point on the tire surface, two independent conditions are required. Usually, the plane of motion, on which the shear stress acts, is known, and the sinkage level or plane is known. These two planes are parallel to the x - y plane and x - z plane, respectively. When the two planes intercept the tire surface, a unique point can be determined. The following gives the method for determining the third coordinate x_0 given the other two coordinates, y_0 and z_0 . To describe the method it is necessary to express the shape functions, N_i , explicitly as follows,

$$N_1 = \frac{1}{4}(1 - \xi)(1 - \eta) \quad (4.99)$$

$$N_2 = \frac{1}{4}(1 - \xi)(1 + \eta) \quad (4.100)$$

$$N_3 = \frac{1}{4}(1 + \xi)(1 + \eta) \quad (4.101)$$

$$N_4 = \frac{1}{4}(1 + \xi)(1 - \eta) \quad (4.102)$$

Since coordinates are interpolated from the nodal coordinates using Eqs. (4.68) to (4.70), the coordinates x_0 , y_0 and z_0 can be expressed as function of ξ and η ,

$$x_0 = A_1 + A_2\xi + A_3\eta + A_4\xi\eta \quad (4.103)$$

$$y_0 = B_1 + B_2\xi + B_3\eta + B_4\xi\eta \quad (4.104)$$

$$z_0 = C_1 + C_2\xi + C_3\eta + C_4\xi\eta \quad (4.105)$$

where x_0 , ξ and η are the unknown variables, and A_i , B_i and C_i have the following form (A_i for example),

$$A_1 = \frac{1}{4}(x_1 + x_2 + x_3 + x_4) \quad (4.106)$$

$$A_2 = -\frac{1}{4}(x_1 + x_2 - x_3 - x_4) \quad (4.107)$$

$$A_3 = -\frac{1}{4}(x_1 - x_2 - x_3 + x_4) \quad (4.108)$$

$$A_4 = \frac{1}{4}(x_1 - x_2 + x_3 - x_4) \quad (4.109)$$

where x_1, x_2, x_3 and x_4 are x nodal coordinates.

Solving ξ from Eq. (4.104), and substituting it into Eq. (4.105) gives the following quadratic equation of η ,

$$p\eta^2 + q\eta + r = 0 \quad (4.110)$$

where

$$\left. \begin{aligned} p &= B_4C_3 - C_4B_3 \\ q &= B_4C_1 - C_4B_1 + B_2C_3 + C_2B_3 + y_0C_4 - z_0B_4 \\ r &= B_2C_1 - C_2B_1 + y_0C_3 - z_0B_2 \end{aligned} \right\} \quad (4.111)$$

Therefore, the two roots of Eq. (4.110) are,

$$\eta_{1,2} = \frac{-q \pm \sqrt{q^2 - 4pr}}{2p} \quad (4.112)$$

Because the two roots are subject to the condition of $-1 \leq \eta \leq +1$, only one of the roots is valid.

Substituting η back into Eq. (4.104), one obtains the value of ξ ,

$$\xi = \frac{y_0 - (B_1 + B_3\eta)}{B_2 + B_4\eta} \quad (4.113)$$

Substituting ξ and η back into Eq. (4.103), the value of x_0 can be determined.

An example of such applications is that if the plane of motion is given, which means the value of z_0 is given, and the ground level is known, which means the value of y_0 is known, the x coordinate of the initial contact point can then be determined.

In the case that the coordinates x_0 and y_0 are related to the contact angle, i.e., $x_0 = -y_0 \tan \theta$, where y_0 is unknown and θ is given, the solution procedure for η , ξ and x_0 becomes,

$$x_0 = -y_0 \tan \theta = A_1 + A_2\xi + A_3\eta + A_4\xi\eta \quad (4.114)$$

$$y_0 = B_1 + B_2\xi + B_3\eta + B_4\xi\eta \quad (4.115)$$

$$z_0 = C_1 + C_2\xi + C_3\eta + C_4\xi\eta \quad (4.116)$$

Substituting y_0 in Eq. (4.115) into Eq. (4.114) gives,

$$0 = D_1 + D_2\xi + D_3\eta + D_4\xi\eta \quad (4.117)$$

where

$$D_i = A_i + B_i \tan \theta \quad i = 1, 2, 3, 4 \quad (4.118)$$

Equation (4.117) together with Eqs. (4.115) and (4.116) has the same form as Eqs. (4.103) to (4.105) except that the x_0 is set to zero and the coefficients, A_i in Eq. (4.103), are replaced by D_i in Eq. (4.118). Therefore, the remaining procedure will be the same.

An example of such applications is that if the plane of motion is given, which means the value of z_0 is given, and the contact angle is known, which means the value of θ is known, the x and y coordinates of the contact point can then be determined.

Based on the formulation for locating the points on the tire surface, the shear displacement can be calculated. During the course of study it was found that at low slip the shear displacement calculated from Eq. (4.98) sometimes is negative, especially when tire deformation is large or the point is on the sidewall. Similar findings were reported by Chen [16], but no explanations were given. In fact, negative shear displacement does exist and all the expressions for calculating shear displacements derived are still valid. Figure 4.19 shows an example of negative shear displacement. At point χ on the sidewall, the radius is R_χ , and it has a forward tangential velocity. Since shear displacement is the displacement relative to the terrain, and is implicitly defined as backward positive, the forward tangential velocity at the point in question is then negative. The shear displacement at χ is the integration of the shear velocity and is negative. It is not necessary that all the shear velocities are negative in order to have negative shear displacement such as the case shown in the example. Sometimes, not all the shear velocities are negative, however, integration of these shear velocities still yields negative shear displacement.

Although the expressions for shear displacement are still valid, Eq. (4.2) for calculating shear stress needs to be modified to suit negative shear displacement. From the physical meaning of shear displacement, Eq. (4.2) is modified as follows,

$$\tau = (c + p \tan \phi)(1 - e^{-\frac{j}{k}}) \quad (\text{if } j \geq 0) \quad (4.119)$$

$$\tau = -(c + p \tan \phi)(1 - e^{\frac{j}{k}}) \quad (\text{if } j < 0) \quad (4.120)$$

4.2.4 Slip-Sinkage Model

The slip-sinkage model for the finite element method is essentially the same as that for the practical model except some special treatment associated with the finite element method.

The differential area dA in Eq. (4.54) is the same as $|d\vec{A}|$ as defined by Eq. (4.86), therefore, Eq. (4.54) for the entire contact area is,

$$\sum_{i=1}^{N_c} \int_{-1}^{+1} \int_{-1}^{+1} p(z_s - z_s^*) |a| d\xi d\eta = 0 \quad (4.121)$$

where N_c is the number of elements in contact with the terrain.

To evaluate the integral in Eq. (4.121), Gauss quadrature is used once again,

$$\begin{aligned} \int_{-1}^{+1} \int_{-1}^{+1} p(z_s - z_s^*) |a| d\xi d\eta \\ = \sum_{k=1}^m \sum_{l=1}^m \left\{ \left[\sum_{i=1}^n N_i(\xi_k, \eta_l) p_i \right] \left[\sum_{i=1}^n N_i(\xi_k, \eta_l) (z_s - z_s^*)_i \right] |a| w_k w_l \right\} \end{aligned} \quad (4.122)$$

The definition of apparent slip-sinkage in the radial direction developed in the previous section is once again adopted for use in the finite element formulation. Since the finite element tire model is an elastic model, the definition for slip-sinkage is further approximated to simplify the problem. Figure 4.18b shows that the slip-sinkage in the plane of motion is defined as the distance between point P and point P' , as described in the previous section. The procedure for calculating the slip-sinkage is outlined below:

- rotating the tire model at sinkage Z about point A so that the slip-sinkage at the wheel center is Z_0
- using the nodal coordinates on surface AP' to determine angle θ
- since P and P' have the same θ , using the procedure outlined in Section 4.2.3 to locate point P .
- subtracting the distance PO from $P'O$ gives the slip-sinkage, z_s .

In this finite element slip-sinkage model, the influence of the change in contact area during slip-sinkage process on the tire geometry is considered small in comparison with that during sinkage process, and is ignored to reduce the computation time.

4.2.5 Incremental-Iterative Scheme and Convergence Criterion

The finite element tire-terrain interaction model involves material nonlinearities as shown in Chapter 2. The equivalent Young's moduli are functions of normal load. The reactions from the terrain are dependent on the tire geometry as described in Chapter 3. This induces geometric nonlinearities, since the load applied will affect the tire geometry in return. Mathematically, the problem can be stated as,

$$\left. \begin{aligned} \vec{d} &= F_1(\vec{x}, \vec{f}_1, \vec{E}) \\ \vec{f}_2 &= F_2(\vec{x}, \vec{d}, \vec{T}_p) \\ \vec{E} &= F_3(\vec{f}_2) \end{aligned} \right\} \quad (4.123)$$

where $\vec{x}$ is the coordinate vector representing the tire geometry before loading, $\vec{f}_1$ is the force vector representing the terrain reactions based on $\vec{x}$, $\vec{d}$ is the displacement vector representing tire deformation due to force $\vec{f}_1$, $\vec{f}_2$ is the force vector representing the terrain reaction based on current tire geometry, $\vec{E}$ represents the Young's moduli and $\vec{T}_p$ represents the terrain parameters.

From Eq. (4.123) it is clear that an iterative scheme is necessary to obtain the deformed shape of the tire for small load. To obtain the deformed shape of the tire for large load or large sinkage, an incremental procedure has to be adopted. In this study the sinkage is considered large, the incremental procedure is therefore used. The increment is selected such that the increase in load due to this increment is small. Within each increment the iterative scheme is followed. The flowchart for the tire-terrain interaction model using the finite element method is illustrated in Fig. 4.20. The procedure can be summarized as follows:

1. reading the input data, including finite element mesh and terrain parameters.
2. incrementing the sinkage or lowering the wheel center by a small amount (one increment), and checking the elements that are in contact with the terrain.
3. calculating stiffness matrix for each element with the Young's moduli all equal to unity.

Figure 4.20 Flowchart of Finite Element Tire-Terrain Interaction Model

4. updating the forces from the terrain using the current geometry and given terrain parameters, and updating the rolling radius $\bar{R}$ using the current geometry and the Young's moduli using the forces just obtained.
5. assembling the global stiffness matrix with the newly updated Young's moduli.
6. applying a fraction of the differential loads between the current loads and those from the previous increment to the tire model to achieve quick and stable convergence.
7. solving the simultaneous linear algebraic equations.
8. checking the convergence for current increment. If convergence is achieved, the slip-sinkage at the wheel centre is determined using work equivalence method and the output for current increment is produced. If not, the loads and new geometry for current iteration are saved, the process goes back to step 4 and continues.
9. checking whether the required sinkage is reached. If yes, the program is terminated. If not, the loads and geometry are saved, and the process goes back to step 2 and continues.

The mathematical representation of step 6 in the above procedure is as follows,

$$\Delta \bar{F}_{i+1} = \Delta \bar{F}_i - \lambda(\Delta \bar{F}_i - \Delta \bar{F}_{i-1}) \quad (4.124)$$

where i is the number of iteration, λ is a positive constant ($0 < \lambda < 1$) and $\Delta \bar{F}_i$ is the differential force vector between two consecutive increments, i.e.,

$$\Delta \bar{F}_i = \bar{F}_k - \bar{F}_{k-1} \quad (4.125)$$

where k is the number of increment.

During the course of study it was found that the value of λ affects convergence significantly, and an inappropriate value of λ may result in the failure of the search. For a relatively rigid tire the value of λ can be chosen around 0.3 to 0.4 to achieve quick convergence. On the other hand, for tire with low inflation pressures, a larger value of λ around 0.7 to 0.8 should be used. It was also found that it is important

to scale the load distribution and maintain its shape using the above-noted method rather than using a non-linear extrapolation scheme that may distort the shape of force distribution and cause divergence.

To terminate the iteration, the following convergence criterion is used,

$$\epsilon = \sqrt{\frac{\sum_{j=1}^N [\Delta(F_{i+1})_j - \Delta(F_i)_j]^2}{\sum_{j=1}^N (F_k)_j (F_k)_j}} \quad (4.126)$$

where ϵ is the overall error, N is the total number of elements in force vector.

If the overall error ϵ in Eq. (4.126) is less than a prescribed value, for instance, 10^{-3} , for two consecutive iterations or more, then convergence is considered to be achieved.

4.2.6 Verification of the Incremental-Iterative Scheme

To verify the numerical solution procedure proposed in Section 4.2.5, a test case which has an analytical solution was designed. Figure 4.21 shows a uniform elastic bar with length equal to L is pressed into deformable soil by an amount of Z_0 . The soil is assumed to have the following characteristic,

$$F_z = s_0 z + s_1 z^2 \quad (4.127)$$

where z is the sinkage of the bar, F_z is the reaction on the bar, s_0 and s_1 are positive constants.

It can be seen that Eq. (4.127) has a hardening property, which is similar to the type of terrain used in this study.

The elastic bar, on the other hand, is assumed to have a softening property, which is similar to the pneumatic tire used in the study. The characteristic for the elastic bar is as follows,

$$F_\delta = b_0 \delta - b_1 \delta^2 \quad (4.128)$$

where δ is the deflection of the elastic bar, F_δ is the force exerted on the bar, b_0 and b_1 are positive constants.

Figure 4.21 Illustration of a Test Case for Verifying the Incremental-Iterative Scheme

For the given conditions shown in Fig. 4.21, the equilibrium sinkage can be obtained by equating Eq. (4.127) to Eq. (4.128) and using the geometric compatibility condition $Z_0 = z + \delta$,

$$z_{1,2} = \frac{-(s_0 + b_0 - 2b_1 Z_0) \pm \sqrt{(s_0 + b_0 - 2b_1 Z_0)^2 - 4(b_1 + s_1)(b_1 Z_0^2 - b_0 Z_0)}}{2(b_1 + s_1)} \quad (4.129)$$

One of the above roots is a valid solution. If assuming $Z_0 = 10$, $s_0 = 150$, $s_1 = 15$, $b_0 = 200$ and $b_1 = 10$, then the sinkage is 4 from Eq. (4.129) and the deflection is 6. If the incremental-iterative scheme is used, the solution accuracy will depend on the number of increment and the accuracy of the solution for each increment. For this test case the allowable error for obtaining the solution for each increment is set at 10^{-9} , and the value of λ is set at 0.6. The following table illustrates the effectiveness of the numerical scheme for this test case,

Table 4.1: Results of Test Case for Verifying the Incremental-Iterative Scheme

(Exact solution: Sinkage = 4; Deflection = 6)

Increment (#)	Sinkage (with iteration)	Relative Error (%)	Sinkage (without iteration)	Relative Error (%)
10	4.0774	1.86	3.6963	-7.5927
20	4.0383	0.9575	3.8568	-3.5791
50	4.0152	0.38	3.9452	-1.3690
100	4.0076	0.19	3.9731	-0.6731
200	4.0038	0.095	3.9867	-0.3336
500	4.0015	0.0375	3.9947	-0.1327
1000	4.0006	0.008	3.9974	-0.0662

Some numerical error should be expected when this incremental-iterative scheme is used for solving finite element tire-terrain interaction problem, since it is not practical to take a large number of increments to achieve higher accuracy. In the finite element solutions the number of increments is usually around 50.

Chapter 5

Experimental Verification of Tire–Terrain Interaction Models

This chapter describes the forced-slip tire tester used in the experimental verification of the two tire–terrain interaction models. Results obtained and discussions on the two models are also presented.

5.1 Forced-Slip Tire Tester

The full-scale forced-slip tire tester built in the Transport Technology Research Laboratory at Carleton University is illustrated in Fig. 5.1.

The tire tester consists of the horizontal motion carriage, the double parallelogram mechanism and the slip pulley–chain system. Horizontal forces on the tire are measured through the two load cells located in front of the carriage frame and at the rear end of the cable. The horizontal movement of the tire is monitored by the shaft encoder mounted on the shaft. The vertical movement of the tire is measured by a potentiometer attached to the double parallelogram mechanism as shown in Fig. 5.1.

Figure 5.2 illustrates a simplified diagram of the forced-slip tire tester for analyzing the mechanics of the tester. As shown in Fig. 5.2, sprocket II and the tire are mounted on the same shaft, while sprocket I and the slip pulley are on the other shaft. Power is

Figure 5.1 Schematic of Forced-Slip Tire Tester

Figure 5.2 Free-Body Diagram of Forced-Slip Tire Tester

transmitted between the two sprockets through the chain. The horizontal movement of the system is restricted by the cable, which is wound around the slip pulley, and there is no slip between the cable and the slip pulley.

It is well known the slip of a wheel is defined as follows,

$$i = 1 - \frac{\text{actual displacement}}{\text{theoretical displacement}} \quad (5.1)$$

In case of the tire tester, the actual displacement of the tire center is equal to the amount of cable length unwound from the slip pulley, $R_1\omega_1 t$. However, if there were no constraints, the tire center should have a displacement of $R_2\omega_2 t$. Therefore, the slip for the tire is,

$$\begin{aligned} i &= 1 - \frac{R_1\omega_1 t}{R_2\omega_2 t} \\ &= 1 - \frac{R_1\omega_1}{R_2\omega_2} \end{aligned} \quad (5.2)$$

where R_1 and R_2 are the radii of the slip pulley and tire, respectively, and ω_1 and ω_2 are the angular velocities for shaft I and shaft II, respectively.

Since two sprockets are chained together, the following relationship exists,

$$\omega_1 r_1 = \omega_2 r_2 \quad (5.3)$$

where r_1 and r_2 are the radii for sprocket I and sprocket II, respectively.

Substituting Eq. (5.3) into Eq. (5.2), one can obtain,

$$i = 1 - \frac{R_1 r_2}{R_2 r_1} \quad (5.4)$$

The ratio of r_2 to r_1 in Eq. (5.4) can be replaced by the gear ratio, η_g , which is the number of teeth of sprocket II divided by the number of teeth of sprocket I, i.e.,

$$i = 1 - \eta_g \frac{R_1}{R_2} \quad (5.5)$$

If the entire tire tester is considered as a free body and is in equilibrium, then the horizontal equilibrium condition is,

$$F_t + T - R_m - F_c = 0 \quad (5.6)$$

where F_t the towing force acting on the carriage, F_c is the cable force, and T and R_m are the thrust and motion resistance of the tire, respectively.

The drawbar pull of the tire can then be calculated as follows,

$$DB = T - R_m = F_c - F_t \quad (5.7)$$

where DB is the drawbar pull.

The thrust T and motion resistance R_m may be estimated using the following equations based on the moment equilibrium analysis for each shaft,

$$\begin{aligned} T &= \frac{F_c R_1}{r_1} \frac{r_2}{R_2} \\ &= F_c (1 - i) \end{aligned} \quad (5.8)$$

$$R_m = T - DB \quad (5.9)$$

It should be pointed out that Eq. (5.8) gives the true value of T for a rigid wheel on hard ground, where R_m is zero. To obtain the true value of T for tire interacting with terrain, the point of application of T , which is not necessarily at the bottom of the tire, must be considered. Furthermore, the tire deformation affects the value of slip, since the actual rolling radius may not be equal to R_2 . This means the theoretical displacement of the tire is difficult to obtain, although the actual displacement can be easily measured.

From Eq. (5.5) it can be seen that by changing the size of slip pulley, R_1 , various slips can be realized. The experiments were carried out at six different slips, 2%, 5%, 10%, 20%, 35% and 50%.

Various tire inflation pressures, 30 psi (206.83 kPa), 20 psi (137.89 kPa) and 10 psi (68.94 kPa), were used in tests to examine the effect of tire deformation on tractive performance.

The data are collected by the computerized data acquisition system outlined in Fig. 5.3.

5.2 Experiments

A series of forced-slip tests were conducted using a Michelin 11.00R16 XL tire on a dry sand. Each type of test was repeated five times to reduce random errors. Figures 5.4 to 5.9 show the results of a set of forced-slip tests, with slip ranging from 2% to 50%.

As can be seen from those diagrams, the starting and ending portions of the test data are generally not reliable since the tire tester was not running in a steady state condition. Therefore, only the data under steady state condition were used in the analysis. It was also observed that some dynamic effects were involved during the tests. This is reflected by the fluctuations shown in these figures. The fluctuations increased with the increase of slip. Slip-stick phenomena, which can best be described as the alternating actions of slip and stick between tire and soil, were observed. This may be the main cause of the dynamic effects. Slip-stick caused the tire and the double parallelogram not to move in a steady manner. A periodic motion was superimposed on the steady movement. This caused the change in the effective vertical loading and affected the horizontal force readings. Therefore, the test data in the steady state range were averaged to filter out the dynamic effects. The processed test data are tabulated in the following tables,

Table 5.1: Measured Thrusts
Inflation Pressure = 30 psi (206.83 kPa)

Slip (%)	Normal Load (N)	Thrusts (N)				
		Test # 1	Test # 2	Test # 3	Test # 4	Test # 5
2	4207.6	450.3	414.1	404.5	398.9	393.4
5	4196.4	474.5	477.9	473.0	470.8	471.0
10	4183.1	712.3	729.4	735.3	730.1	745.9
20	4137.0	1124.4	1175.8	1178.3	1173.0	1188.8
35	4061.9	1456.2	1470.5	1474.8	1479.4	1475.8
50	4029.1	1645.8	1650.0	1642.9	1633.6	1641.1

Tire Tractive Performance in Sand at 2% Slip Inflation Pressure = 30 psi (206.83 kPa)

Figure 5.4 Sample of Measured Forced-Slip Tire Test Data at 2% Slip

Tire Tractive Performance in Sand at 5% Slip Inflation Pressure = 30 psi (206.83 kPa)

Figure 5.5 Sample of Measured Forced-Slip Tire Test Data at 5% Slip

Tire Tractive Performance in Sand at 10% Slip Inflation Pressure = 30 psi (206.83 kPa)

Figure 5.6 Sample of Measured Forced-Slip Tire Test Data at 10% Slip

Tire Tractive Performance in Sand at 20% Slip Inflation Pressure = 30 psi (206.83 kPa)

Figure 5.7 Sample of Measured Forced-Slip Tire Test Data at 20% Slip

Tire Tractive Performance in Sand at 35% Slip Inflation Pressure = 30 psi (206.83 kPa)

Figure 5.8 Sample of Measured Forced-Slip Tire Test Data at 35% Slip

Tire Tractive Performance in Sand at 50% Slip Inflation Pressure = 30 psi (206.83 kPa)

Figure 5.9 Sample of Measured Forced-Slip Tire Test Data at 50% Slip

Table 5.2: Measured Resistances
Inflation Pressure = 30 psi (206.83 kPa)

Slip (%)	Normal Load (N)	Resistances (N)				
		Test # 1	Test # 2	Test # 3	Test # 4	Test # 5
2	4207.6	970.2	951.6	940.2	950.1	949.2
5	4196.4	919.4	916.4	912.5	910.2	911.1
10	4183.1	885.6	885.1	894.2	878.0	895.2
20	4137.0	911.9	926.0	915.8	898.9	913.8
35	4061.9	1078.7	1107.5	1123.3	1129.2	1134.2
50	4029.1	1252.5	1271.5	1234.4	1259.7	1247.0

Table 5.3: Measured Drawbar Pulls
Inflation Pressure = 30 psi (206.83 kPa)

Slip (%)	Normal Load (N)	Drawbar Pulls (N)				
		Test # 1	Test # 2	Test # 3	Test # 4	Test # 5
2	4207.6	-520.0	-537.5	-535.7	-551.1	555.8
5	4196.4	-444.9	-438.4	-439.5	-439.4	440.1
10	4183.1	-173.3	-155.7	-158.9	148.0	149.3
20	4137.0	212.5	249.8	262.5	274.1	275.0
35	4061.9	377.5	363.0	351.4	350.2	341.6
50	4029.1	393.3	378.4	408.5	373.9	394.1

Table 5.4: Measured Total Sinkages

Inflation Pressure = 30 psi (206.83 kPa)

Slip (%)	Normal Load (N)	Total Sinkages (cm)				
		Test # 1	Test # 2	Test # 3	Test # 4	Test # 5
2	4207.6	7.58	7.51	7.33	6.99	6.93
5	4196.4	6.74	6.89	7.54	6.83	6.86
10	4183.1	7.18	7.96	7.48	7.34	7.50
20	4137.0	9.33	8.14	8.53	8.60	8.20
35	4061.9	10.09	10.25	10.11	10.56	9.34
50	4029.1	11.98	12.39	12.04	12.37	11.93

Table 5.5: Measured Thrusts

Inflation Pressure = 20 psi (137.89 kPa)

Slip (%)	Normal Load (N)	Thrusts (N)				
		Test # 1	Test # 2	Test # 3	Test # 4	Test # 5
2	4207.6	479.1	399.2	377.2	349.2	354.1
5	4196.4	443.3	444.0	532.7	442.5	443.3
10	4183.1	693.7	705.6	702.1	702.4	704.3
20	4137.0	1125.9	1171.7	1179.6	1177.7	1186.3
35	4061.9	1467.7	1486.6	1502.9	1487.8	1493.8
50	4029.1	1601.8	1598.7	1598.5	1594.6	1598.7

Table 5.6: Measured Resistances
Inflation Pressure = 20 psi (137.89 kPa)

Slip (%)	Normal Load (N)	Resistances (N)				
		Test # 1	Test # 2	Test # 3	Test # 4	Test # 5
2	4207.6	883.5	901.5	895.3	884.4	890.7
5	4196.4	882.8	884.1	974.0	879.3	887.0
10	4183.1	871.0	855.6	855.8	854.6	852.4
20	4137.0	906.0	897.0	900.5	890.0	891.8
35	4061.9	1084.4	1110.7	1118.2	1119.4	1115.2
50	4029.1	1199.7	1186.4	1186.1	1177.8	1188.1

Table 5.7: Measured Drawbar Pulls
Inflation Pressure = 20 psi (137.89 kPa)

Slip (%)	Normal Load (N)	Drawbar Pulls (N)				
		Test # 1	Test # 2	Test # 3	Test # 4	Test # 5
2	4207.6	-404.3	-502.4	-518.0	-535.2	-536.6
5	4196.4	-439.5	-440.1	-441.2	-436.8	-443.7
10	4183.1	-177.2	-150.0	-153.7	-152.2	-148.1
20	4137.0	219.9	274.7	279.1	287.7	294.5
35	4061.9	383.4	376.0	384.7	368.4	378.5
50	4029.1	402.1	412.3	412.4	416.8	410.7

Table 5.8: Measured Total Sinkages

Inflation Pressure = 20 psi (137.89 kPa)

Slip (%)	Normal Load (N)	Total Sinkages (cm)				
		Test # 1	Test # 2	Test # 3	Test # 4	Test # 5
2	4207.6	8.00	6.72	7.11	7.01	7.11
5	4196.4	6.99	6.66	7.27	7.23	7.10
10	4183.1	7.38	7.31	7.22	7.05	7.15
20	4137.0	7.87	8.57	8.43	8.17	8.57
35	4061.9	9.99	9.80	10.45	10.39	10.21
50	4029.1	12.32	11.94	11.93	11.98	12.23

Table 5.9: Measured Thrusts

Inflation Pressure = 10 psi (68.94 kPa)

Slip (%)	Normal Load (N)	Thrusts (N)				
		Test # 1	Test # 2	Test # 3	Test # 4	Test # 5
2	4207.6	401.9	286.2	261.8	250.9	229.2
5	4196.4	326.0	323.4	319.3	317.4	320.4
10	4183.1	627.0	626.8	624.6	628.5	630.4
20	4137.0	1131.9	1181.4	1084.2	1186.0	1189.4
35	4061.9	1504.3	1520.8	1521.5	1530.8	1529.9
50	4029.1	1679.4	1683.6	1691.3	1694.0	1687.9

Table 5.10: Measured Resistances

Inflation Pressure = 10 psi (68.94 kPa)

Slip (%)	Normal Load (N)	Resistances (N)				
		Test # 1	Test # 2	Test # 3	Test # 4	Test # 5
2	4207.6	794.3	807.5	798.1	788.3	783.4
5	4196.4	767.4	756.8	757.0	757.9	756.2
10	4183.1	728.2	722.3	710.9	718.8	712.2
20	4137.0	778.6	757.8	672.0	750.6	751.4
35	4061.9	997.3	1015.3	1018.0	1024.4	1023.0
50	4029.1	1170.8	1172.7	1177.3	1182.7	1176.4

Table 5.11: Measured Drawbar Pulls

Inflation Pressure = 10 psi (68.94 kPa)

Slip (%)	Normal Load (N)	Drawbar Pulls (N)				
		Test # 1	Test # 2	Test # 3	Test # 4	Test # 5
2	4207.6	-392.4	-521.4	-536.4	-537.4	-554.1
5	4196.4	-441.4	-433.4	-437.8	-440.5	-435.8
10	4183.1	-101.2	-95.5	-86.2	-90.4	-81.8
20	4137.0	353.3	423.6	412.2	435.4	438.0
35	4061.9	506.9	505.5	503.5	506.4	506.9
50	4029.1	508.6	510.9	514.1	511.3	511.5

Table 5.12: Measured Total Sinkages
Inflation Pressure = 10 psi (68.94 kPa)

Slip (%)	Normal Load (N)	Total Sinkages (cm)				
		Test # 1	Test # 2	Test # 3	Test # 4	Test # 5
2	4207.6	7.94	7.17	7.09	6.98	7.23
5	4196.4	6.65	6.64	6.56	6.66	6.79
10	4183.1	6.90	7.01	7.01	7.33	7.01
20	4137.0	9.20	8.15	8.21	8.13	8.45
35	4061.9	9.58	11.13	10.56	10.50	11.02
50	4029.1	12.49	12.57	12.86	13.01	12.69

As discussed in the previous section, the thrust and motion resistance are estimated from the cable force F_c . Only drawbar pull can be measured indirectly, which can be calculated from Eq. (5.7). The direct measurements are F_c and F_t . It was found during the test that F_c and F_t are close. Thus, any errors in F_c and F_t may cause very large relative error in drawbar pull.

More generally if the numbers A and B have errors of E_a and E_b , respectively, and they are denoted as $A \pm E_a$ and $B \pm E_b$, then the difference between the two numbers, C , will have an error of $E_a + E_b$, or $C \pm (E_a + E_b)$. Take one set of test data [20% slip, 30 psi (206.83 kPa) inflation pressure] as an example, the average F_c is 1403.4 N and the average F_t is 1190.3 N, the average DB is then 213.1 N. It was estimated that the data acquisition system has a measurement error of 6.7 N due to the resolutions of load cells and A/D converter. Assuming F_c and F_t have the same measurement errors of 6.7 N, then the drawbar pull DB will have an error of 13.4 N. The relative error of DB is 6.26%, while F_c and F_t only have 0.475% and 0.56%, respectively. This factor should be kept in mind when comparing the test data with the numerical predictions.

5.3 Numerical Examples

The finite element mesh of the tire model used in the analysis of tire terrain interaction is shown in Fig. 5.10. Due to the symmetry in loading conditions about the plane of motion of the tire, only half of the tire is modeled. The number of elements in the mesh is 588 and the number of nodes is 1260. Parametric studies were conducted and it was found that this type of mesh is suitable for the analysis. The following tables give the numerical simulation results using the model.

Table 5.13: Finite Element Simulation Results

Inflation Pressure = 30 psi (206.83 kPa)

Slip (%)	Normal Load (N)	Thrust (N)	Resistance (N)	Drawbar (N)	Sinkage (cm)	Slip-sinkage (cm)
2	4207.6	130.3	618.3	-488	6.84	1.56
5	4196.4	480	701.5	-221.5	6.7	2.55
10	4183.1	887.5	851.3	36.2	6.57	4.1
20	4137.0	1298.5	1028.2	270.3	6.29	6.69
35	4061.9	1513.0	1145.9	367.1	5.98	9.25
50	4029.1	1596.8	1177.5	419.3	5.76	10.84

Table 5.14: Finite Element Simulation Results

Inflation Pressure = 20 psi (137.89 kPa)

Slip (%)	Normal Load (N)	Thrust (N)	Resistance (N)	Drawbar (N)	Sinkage (cm)	Slip-sinkage (cm)
2	4207.6	0.1	574.9	-574.8	6.70	1.34
5	4196.4	444.3	671.2	-226.9	6.58	2.38
10	4183.1	880.4	815.8	64.6	6.46	3.87
20	4137.0	1301	1008.9	292.1	6.21	6.47
35	4061.9	1517.4	1134.9	382.5	5.92	9.11
50	4029.1	1597.9	1177.9	420.0	5.72	10.76

Michelin 11.00R16 XL

Total Nodes: 1260 Total Elements: 588
Transport Technology Research Laboratory
Carleton University, Ottawa, CANADA

**Figure 5.10 Finite Element Tire Model for
Simulating Tire-Terrain Interaction**

Table 5.15: Finite Element Simulation Results

Inflation Pressure = 10 psi (68.94 kPa)

Slip (%)	Normal Load (N)	Thrust (N)	Resistance (N)	Drawbar (N)	Sinkage (cm)	Slip-sinkage (cm)
2	4207.6	—	—	—	—	—
5	4196.4	220.8	467.9	-247.1	5.98	1.38
10	4183.1	810.2	655.8	154.4	5.89	2.99
20	4137.0	1334.5	861.8	472.7	5.71	5.47
35	4061.9	1544.3	1069	475.3	5.57	9.11
50	4029.1	1610.2	1157.7	452.5	5.43	10.30

The numerical simulation results using the practical tire-terrain interaction model are tabulated in the following table. Calculations show that for the given normal loads and tire inflation pressures, the criterion for elastic mode is not satisfied, and the tire is only operating in the rigid mode.

Table 5.16: Predicted Performance Using the Practical Model

Inflation Pressures = 10 psi (206.83 kPa), 20 psi (137.89 kPa), 30 psi (68.94 kPa)

Slip (%)	Normal Load (N)	Thrust (N)	Resistance (N)	Drawbar (N)	Sinkage (cm)	Slip-sinkage (cm)
2	4207.6	468	828	-360	6.34	3.25
5	4196.4	716	975	-209	6.22	4.36
10	4183.1	1010	1004	6	5.95	5.81
20	4137.0	1346	1106	240	5.6	8.03
35	4061.9	1543	1195	348	5.29	10.32
50	4029.1	1616	1256	360	5.12	11.92

The variations in normal loads in the above tables are due to the weight differentials in the slip pulleys. The results from forced-slip tire tests and the numerical predictions from two tire-terrain interaction models are plotted in Figs. 5.11 to 5.22.

The experimental data shown in Tables 5.1 to 5.12 are averaged and compared

Michelin 11.00R16 XL
Inflation Pressure = 30 psi (206.83 kPa)
on Sand

Figure 5.11 Measured and Predicted Thrust at 30 psi (206.83 kPa)

Michelin 11.00R16 XL
Inflation Pressure = 30 psi (206.83 kPa)
on Sand

**Figure 5.12 Measured and Predicted
Motion Resistance at 30 psi (206.83 kPa)**

Michelin 11.00R16 XL
Inflation Pressure = 30 psi (206.83 kPa)
on Sand

**Figure 5.13 Measured and Predicted
Drawbar Pull at 30 psi (206.83 kPa)**

Michelin 11.00R16 XL
Inflation Pressure = 30 psi (206.83 kPa)
on Sand

**Figure 5.14 Measured and Predicted
Total Sinkage at 30 psi (206.83 kPa)**

Michelin 11.00R16 XL
Inflation Pressure = 20 psi (137.89 kPa)
on Sand

Figure 5.15 Measured and Predicted Thrust at 20 psi (137.89 kPa)

Michelin 11.00R16 XL
Inflation Pressure = 20 psi (137.89 kPa)
on Sand

Figure 5.16 Measured and Predicted
Motion Resistance at 20 psi (137.89 kPa)

Figure 5.17 Measured and Predicted
Drawbar Pull at 20 psi (137.89 kPa)

**Figure 5.18 Measured and Predicted
Total Sinkage at 20 psi (137.89 kPa)**

Michelin 11.60R16 XL
Inflation Pressure = 10 psi (68.94 kPa)
on Sand

Figure 5.19 Measured and Predicted Thrust at 10 psi (68.94 kPa)

Michelin 11.00R16 XL
Inflation Pressure = 10 psi (68.94 kPa)
on Sand

Figure 5.20 Measured and Predicted
Motion Resistance at 10 psi (68.94 kPa)

Michelin 11.00R16 XL
Inflation Pressure = 10 psi (68.94 kPa)
on Sand

**Figure 5.21 Measured and Predicted
Drawbar Pull at 10 psi (68.94 kPa)**

Michelin 11.00R16 XL
Inflation Pressure = 10 psi (68.94 kPa)
on Sand

**Figure 5.22 Measured and Predicted
Total Sinkage at 10 psi (68.94 kPa)**

with the predictions from both the finite element model and the practical model. Relative errors of these two prediction models with respect to the measured data are tabulated below.

Table 5.17: Comparison of Predicted and Measured Thrusts

Inflation Pressure = 30 psi (206.83 kPa)

Slip (%)	Measured Thrust (N)	P. M. Thrust (N)	Relative Error (%)	FEM Thrust (N)	Relative Error (%)
2	412.2	468	13.5	130.3	-68.4
5	473.4	716	51.2	480	1.4
10	730.6	1010	38.2	887.5	21.5
20	1168.1	1346	15.2	1298.5	11.2
35	1471.3	1543	4.9	1513.0	2.8
50	1642.7	1616	-1.6	1596.8	-2.8

Table 5.18: Comparison of Predicted and Measured Resistances

Inflation Pressure = 30 psi (206.83 kPa)

Slip (%)	Measured Resist. (N)	P. M. Resist. (N)	Relative Error (%)	FEM Resist. (N)	Relative Error (%)
2	952.3	828	-13.1	618.3	-35.1
5	913.9	975	6.7	701.5	-23.2
10	887.6	1004	13.1	851.3	-4.1
20	913.3	1106	21.1	1028.2	12.6
35	1114.6	1195	7.2	1145.9	2.8
50	1253.0	1256	0.2	1177.5	-6.0

Note: P.M. is short for Practical Model

Table 5.19: Comparison of Predicted and Measured Drawbar Pulls

Inflation Pressure = 30 psi (206.83 kPa)

Slip (%)	Measured Drawbar (N)	P. M. Drawbar (N)	Relative Error (%)	FEM Drawbar (N)	Relative Error (%)
2	-540.0	-360	33.3	-488	9.6
5	-440.5	-209	52.6	-221.5	49.7
10	-157.0	6	103.8	36.2	123.1
20	254.8	240	5.8	270.3	6.1
35	356.7	348	-2.5	367.1	2.9
50	389.6	360	-7.6	419.3	7.6

Table 5.20: Comparison of Predicted and Measured Total Sinkages

Inflation Pressure = 30 psi (206.83 kPa)

Slip (%)	Measured Sinkage (cm)	P. M. Sinkage (cm)	Relative Error (%)	FEM Sinkage (cm)	Relative Error (%)
2	7.77	9.59	23.4	8.4	8.1
5	6.97	10.58	51.7	9.25	32.7
10	7.89	11.76	49.1	10.67	35.2
20	8.56	13.63	59.2	12.98	51.6
35	10.07	15.61	55.0	15.23	51.3
50	12.14	17.04	40.3	16.6	36.7

Table 5.21: Comparison of Predicted and Measured Thrusts

Inflation Pressure = 20 psi (137.89 kPa)

Slip (%)	Measured Thrust (N)	P. M. Thrust (N)	Relative Error (%)	FEM Thrust (N)	Relative Error (%)
2	391.8	468	19.5	0.1	-100
5	461.2	716	55.4	444.3	-3.6
10	701.6	1010	42.9	880.4	25.5
20	1168.2	1346	15.2	1301	11.4
35	1487.8	1543	3.7	1517.4	19.9
50	1598.5	1616	1.1	1597.9	0.00

Table 5.22: Comparison of Predicted and Measured Resistances

Inflation Pressure = 20 psi (137.89 kPa)

Slip (%)	Measured Resist. (N)	P. M. Resist. (N)	Relative Error (%)	FEM Resist. (N)	Relative Error (%)
2	891.1	828	-7.1	574.9	-34.5
5	901.4	975	8.2	671.2	-25.5
10	857.9	1004	17.0	815.8	-4.9
20	897.1	1106	23.3	1008.9	12.5
35	1109.6	1195	7.7	1134.9	2.3
50	1187.6	1256	5.8	1177.9	-0.8

Table 5.23: Comparison of Predicted and Measured Drawbar Pulls

Inflation Pressure = 20 psi (137.89 kPa)

Slip (%)	Measured Drawbar (N)	P. M. Drawbar (N)	Relative Error (%)	FEM Drawbar (N)	Relative Error (%)
2	-499.3	-360	27.9	-574.8	-15.1
5	-440.3	-209	52.5	-226.9	48.5
10	-156.2	6	103.8	64.6	141.3
20	271.2	240	-11.5	292.1	7.7
35	378.2	348	-8	382.5	1.1
50	410.9	360	-12.4	420.0	2.2

Table 5.24: Comparison of Predicted and Measured Total Sinkages

Inflation Pressure = 20 psi (137.89 kPa)

Slip (%)	Measured Sinkage (cm)	P. M. Sinkage (cm)	Relative Error (%)	FEM Sinkage (cm)	Relative Error (%)
2	7.19	9.59	33.3	8.04	11.9
5	7.11	10.58	50.1	8.96	27.0
10	7.22	11.76	62.9	10.33	43.0
20	8.36	13.63	63.1	12.68	51.7
35	10.17	15.61	53.5	15.03	47.8
50	12.08	17.04	41.1	16.48	36.4

Table 5.25: Comparison of Predicted and Measured Thrusts

Inflation Pressure = 10 psi (68.94 kPa)

Slip (%)	Measured Thrust (N)	P. M. Thrust (N)	Relative Error (%)	FEM Thrust (N)	Relative Error (%)
2	286.0	468	63.6	—	—
5	321.3	716	122.8	220.8	-31.3
10	627.5	1010	61	810.2	29.1
20	1154.6	1346	16.6	1334.5	15.6
35	1521.5	1543	1.4	1544.3	1.5
50	1687.2	1616	-4.2	1610.2	-4.6

Table 5.26: Comparison of Predicted and Measured Resistances

Inflation Pressure = 10 psi (68.94 kPa)

Slip (%)	Measured Resist. (N)	P. M. Resist. (N)	Relative Error (%)	FEM Resist. (N)	Relative Error (%)
2	794.3	828	4.2	—	—
5	759.1	975	28.5	467.9	-38.4
10	718.5	1004	39.7	655.8	-8.7
20	742.1	1106	49.0	861.8	16.1
35	1015.6	1195	17.7	1069	5.3
50	1176	1256	6.8	1157.7	-1.6

Table 5.27: Comparison of Predicted and Measured Drawbar Pulls

Inflation Pressure = 10 psi (68.94 kPa)

Slip (%)	Measured Drawbar (N)	P. M. Drawbar (N)	Relative Error (%)	FEM Drawbar (N)	Relative Error (%)
2	-508.3	-360	29.2	---	---
5	-437.8	-209	52.3	-247.1	43.6
10	-91.0	6	106.6	154.4	269.5
20	412.5	240	-41.2	472.7	14.6
35	505.8	348	-31.2	475.3	-6.0
50	511.3	360	-29.6	452.5	-11.5

Table 5.28: Comparison of Predicted and Measured Total Sinkages

Inflation Pressure = 10 psi (68.94 kPa)

Slip (%)	Measured Sinkage (cm)	P. M. Sinkage (cm)	Relative Error (%)	FEM Sinkage (cm)	Relative Error (%)
2	7.28	9.59	31.7	---	---
5	6.66	10.58	58.8	7.36	10.4
10	7.05	11.76	66.9	8.88	25.8
20	8.43	13.63	61.7	11.18	32.6
35	10.56	15.61	47.8	14.68	39.0
50	12.74	17.04	33.7	15.73	23.5

It can be seen that relative errors in Tables 5.17 to 5.28 vary considerably. The relative errors of practical model vary from 0.2% minimum to 122.8% maximum, and those of finite element model vary from 0.0% to 269.5%. The huge relative errors (greater than 100%) only occur when the absolute values of measured drawbar pulls are small or near zero and the differences between the predictions and measured data are large. It is seen that point-by-point comparison may not reflect completely the correlation of the two curves. It is therefore necessary to select a standard way of comparing curves. For this purpose, an expression for a regression coefficient is introduced,

$$R_c = 1 - \sqrt{\frac{\sum (y_p - y_m)^2}{\sum y_m^2}} \quad (5.10)$$

where y_p and y_m are predicted and measured values, respectively.

The following table gives the regression coefficients for all the curves.

Table 5.29: Regression Coefficients for All Fitted Performance Curves

Tire Inflation Pressure	Tractive Performance Performance	Regression Coefficients			
		Practical Model	FEM Model	Comparison of Models	
				Practical	FEM
30 psi (206.83 kPa)	Thrust	0.842	0.868	better	better
	Resistance	0.889	0.830		
	Drawbar Pull	0.635	0.677		better
	Total Sinkage	0.524	0.599		better
20 psi (137.89 kPa)	Thrust	0.830	0.829	better	better
	Resistance	0.878	0.83	better	
	Drawbar Pull	0.651	0.657		
	Total Sinkage	0.494	0.6		
10 psi (68.94 kPa)	Thrust	0.778	0.892		better
	Resistance	0.728	0.838		better
	Drawbar Pull	0.604	0.656		better
	Total Sinkage	0.502	0.709		better

From Table 5.29 it is clear that the finite element model is generally better than the practical model (9 cases against 3 cases). When the tire inflation pressure is low, all predictions from the finite element model are better than those from the practical model. Regression coefficients are meaningful only in relative terms. In other words, regression coefficients for one parameter should not be compared with those for the others, since the magnitudes for the different parameters are different.

Comparisons between the predicted and measured tractive performance at 20% slip are summarized in Table 5.30. For this particular slip, the predictions from the finite element model are much better than those from the practical model (11 cases against 1 case).

As discussed in Chapter 4, the pressure distribution in the contact area affects the tractive performance significantly. It appears that both proposed models give reasonable prediction for pressure distributions. Figure 5.23 shows the normal pressure distributions from the practical model. It can be seen that as the slip increases, the contact angle becomes larger, the value of the maximum pressure decreases, and the position of the maximum pressure shifts forward. Experimental observations of this were reported by Oida et al [7] (see Figs. F.1 and F.2 in Appendix F for details). The position change of the maximum pressure with the slip and the effects of slip-sinkage are the reasons why the motion resistance increases with the slip. Features similar to those normal pressure distributions are also observed for shear stress distributions as shown in Fig. 5.24. However, the value of the maximum shear stress increases with the slip. These were also reported by these researchers (see Fig. F.1). Figure 5.25 shows that the maximum shear stress and normal pressure do not occur at the same angular position. The difference between the maxima is about 5.4° . This prediction is contradictory to what they observed. According to the experiment by Oida et al [7] the maximum shear stress is in front of the maximum normal pressure not behind. However, different observations about the positions of the maximum normal pressure and shear stress were reported by Wong [35]. They found that the maximum shear stress is always behind the maximum normal pressure. Certainly, further investigation in this regard is necessary.

Predictions for the pressure and shear stress distributions from the finite element

Table 5.30: Comparison of Predicted and Measured Tractive Performance at 20% Slip

Tire Inflation Pressure	Tractive Performance Parameter	Measured	Practical Model	Relative Error (%)	FEM Model	Relative Error (%)	Comparison of Models	
							Practical	FEM
30 psi (206.83 kPa)	Thrust, N	1168.1	1346	15.2	1298.5	11.2		Better
	Resistance, N	913.3	1106	21.1	1028.2	12.6		Better
	Drawbar Pull, N	254.8	240	5.8	270.3	6.06	Better	
	Total Sinkage, cm	8.56	13.63	59.2	12.98	51.6		Better
20 psi (137.89 kPa)	Thrust, N	1168.2	1346	15.2	1301	11.4		Better
	Resistance, N	897.1	1106	23.3	1008.9	12.5		Better
	Drawbar Pull, N	271.2	240	-11.5	292.1	7.7		Better
	Total Sinkage, cm	8.36	13.63	63.1	12.68	51.7		Better
10 psi (68.94 kPa)	Thrust, N	1154.6	1346	16.6	1334.5	15.6		Better
	Resistance, N	742.08	1106	49.0	861.8	16.1		Better
	Drawbar Pull, N	412.5	240	-41.2	472.7	14.6		Better
	Total Sinkage, cm	8.43	13.63	61.7	11.18	32.6		Better

Michelin 11.00R16 XL On Sand

Figure 5.23 Normal Pressure Distributions from Practical Model for Various Slips

Michelin 11.00R16 XL
On Sand

Figure 5.24 Shear Stress Distributions from
Practical Model for Various Slips

Michelin 11.00R16 XL On Sand

Figure 5.25 Illustration of Normal Pressure and Shear Stress Distributions from Practical Model for a Given Slip

model are similar to those from the practical model. Since the tire surface in the finite element model is three-dimensional, the pressure and shear stress on the tire surface are functions of three variables, namely the three coordinates. It is difficult to realistically present a function of three variables in a three-dimensional space. Therefore, projections of pressure and shear stress distributions are presented. Figures 5.26, 5.27 and 5.28 illustrate an example of the pressure distribution projected on the x - y plane (side view), x - z plane (top view) and y - z plane (front view), respectively.

The prediction for the tractive performance from the finite element model is in general better than that from the practical model. In comparison with the test data, both models tend to over-estimate the thrust and to under-estimate the resistance in low slip range. From the diagrams for motion resistance (Figs. 5.12, 5.16 and 5.18), it can be seen that the measured resistance varies from high through low to very high as the slip increases. On the other hand, the resistance predicted increases with slip monotonically. This discrepancy may be due to two reasons. One is when the tire operating at low slips during the experiment the tire may actually be skidding. This is due to the uncertainty in using the outer diameter of the undeformed tire for calculating the slip. The other reason is that the method for calculating shear displacement in the two tire-terrain interaction models does not take into account the movement of the soil when the tire is rolling.

As mentioned earlier, slip-sinkage has seldom been incorporated into tire-terrain interaction model. Figures 5.29 to 5.31 demonstrate the importance of incorporating slip-sinkage into a tire-terrain interaction model. They show that there is a significant difference between the tractive performance predicted by the practical model with slip-sinkage considered and that without including slip-sinkage. It can be seen that without incorporating slip-sinkage into a model it will not produce close prediction of the tractive performance of a tire over frictional terrain, where slip-sinkage is significant.

Although the practical model predicts the tractive performance reasonably well for relatively high inflation pressure, it fails to give a good prediction for low inflation pressure. The reason is that the criterion used for determining the operating mode may not be adequate. For the three inflation pressures used, the practical model

Pressure distribution

154

Transport Technology Research Laboratory
Carleton University, Ottawa, CANADA

Figure 5.26 Normal Pressure Distribution from FEM Model: Side View

Pressure distribution

155

Transport Technology Research Laboratory
Carleton University, Ottawa, CANADA

Figure 5.27 Normal Pressure Distribution from FEM Model: Top View

Pressure distribution

Transport Technology Research Laboratory
Carleton University, Ottawa, CANADA

Figure 5.28 Normal Pressure Distribution from FEM Model: Front View

Michelin 11.00R16 XL
Inflation Pressure = 30 psi (206.83 kPa)
on Sand

Figure 5.29 Effect of Slip-Sinkage on Thrust for Practical Model

Michelin 11.00R16 XL
Inflation Pressure = 30 psi (206.83 kPa)
on Sand

Figure 5.30 Effect of Slip-Sinkage on
Motion Resistance for Practical Model

Michelin 11.00R16 XL
Inflation Pressure = 30 psi (206.83 kPa)
on Sand

Figure 5.31 Effect of Slip-Sinkage on Drawbar Pull for Practical Model

predicts that the tire is always operating in the rigid mode. Therefore, there is no change in the tire geometry, and tire behaves like a rigid wheel. Using the finite element model, the tire is considered to be an elastic body. Therefore, the finite element model gives better prediction of the contact geometry than the practical model. Consequently, the predictions from the finite element model are generally better than those from the practical model for all inflation pressures.

Although the finite element model gives good predictions for thrust, resistance and drawbar pull, it over-estimates the tire sinkage as shown in Figs 5.14, 5.18 and 5.22. This may be due to many reasons. First, the assumption that the slip-sinkage will not affect the tire geometry is not realistic. In fact, the contact area after the slip-sinkage is larger than that before slip-sinkage occurs. Therefore, there must be some changes in the contact geometry due to slip-sinkage. However, the effect of this change on the tractive performance is not considered because it demands much more computing power than currently available. Second, the details of the complex lug pattern has not been taken into account. Third, using two Young's moduli to approximate the complex material properties of the tire carcass may not be sufficient. The method for deriving the ratio of Young's moduli may also require further improvement. As shown in Figs. 2.10 and 2.11 the predicted contact area-deflection relationship from the finite element model does not fit the test data as well as those for load deflection curves. Finally, the systematic errors in all experimental measurements also contribute to the discrepancy between the measured tractive performance and the predictions from the two tire-terrain interaction models.

Chapter 6

Conclusions and Recommendations

6.1 Conclusions

Based on the results of the study of tire-terrain interaction described in this thesis, the following conclusions can be drawn:

- Two tire-terrain interaction models, one based on the finite element technique and the other based on a semi-empirical approach, referred to as the “practical model”, were developed for predicting tractive performance of pneumatic tires operating over deformable terrain. Reasonably good correlations were achieved between the measured performance obtained using a forced-slip tire tester and the predicted ones obtained using both models.
- In general the finite element tire-terrain interaction model is better than the practical tire-terrain interaction model in that it gives a more realistic prediction of the tire-terrain contact geometry. However, the computation time using the finite element model is approximately 100 times longer than that using the practical model. Consequently, the selection of an appropriate model will be dependent upon the requirements of a given task.
- For tires with high inflation pressure and operating over relatively soft terrain, the practical tire-terrain interaction model gives good predictions of tractive

performance at high slips (greater than 20%), and reasonable predictions at low slips (less than 20%).

- For tires with low inflation pressure and operating over relatively firm terrain, the use of the finite element tire-terrain interaction model is necessary to achieve a more accurate prediction of the tractive performance. The main advantage of using the finite element method is that it provides a more realistic prediction of the tire-terrain contact geometry.
- Slip-sinkage is a very important factor in modeling tire-terrain interaction on frictional terrain. It is shown that the tire-terrain interaction model without incorporating slip-sinkage does not provide adequate prediction of tractive performance (see Figs. 5.29, 5.30 and 5.31).
- To incorporate slip-sinkage into tire-terrain interaction models, an additional condition or assumption is required to determine the slip-sinkage at the wheel center. This can be accomplished by assuming (a) either there exist two vertical equilibrium positions, one before and one after slip-sinkage takes place, for the entire process, or (b) there exists one equilibrium position with an assumed relationship between the sinkage and slip-sinkage at the wheel center.
- Studies show that between the two approaches for incorporating slip-sinkage effects into tire-terrain interaction models, the one-equilibrium-position approach with work equivalence assumption yields better predictions than the other, and has a wider range of applications. The two-equilibrium-position approach is restricted to two-dimensional problems, while the work equivalence approach is suitable for both two- and three-dimensional problems.
- The normal pressure and shear stress distributions predicted using both the practical and the finite element tire-terrain interaction models show similar characteristics as those measured and reported in the literature [6, 7]. For instance, the location of the maximum pressure shifting forward with increase in slip is consistent with experimental observations.

- To ensure the convergence of the numerical procedure using the finite element tire-terrain interaction model, a simple and yet effect numerical search scheme was developed and validated. This is an improvement over the previous finite element tire-terrain interaction model [16].
- A more concise analytical method was developed for calculating shear stress under continuously varying normal pressure. It is shown that the new method is more efficient and accurate than the numerical method developed previously [35].
- It is found that the bevameter technique provides a reasonably good method for measuring and characterizing the mechanical behaviour of deformable terrain. The three semi-empirical relationships measured by the bevameter, namely, the relationship between normal pressure and sinkage, the relationship between shear stress and shear displacement, and the relationship between slip-sinkage and shear displacement, provide an adequate representation of terrain behaviour for vehicle mobility study.
- An improved method for searching for the equivalent Young's moduli was developed [16, 25]. It is shown that if the improved method is used, the equivalent Young's moduli can be obtained more efficiently and accurately.

6.2 Recommendations

During the course of study it has been found that certain aspects of the tire-terrain interaction models presented in this thesis require further improvements.

- Further studies of the slip-sinkage phenomenon and its characterization are recommended. The current method follows one-equilibrium-position approach with work equivalence assumption to determine the sinkage and slip-sinkage at the wheel center. Further verification of the work equivalence assumption is desirable.

- The finite element tire model currently utilizes a type of 8-node isoparametric element. This type of element approximates the surface of the tire carcass with small facets, and can not model curvature. To obtain more realistic representation of the geometry of the tire carcass, a great deal more elements are required, which requires much longer time to complete the computation. The 20-node isoparametric element, on the other hand, is capable of modeling curvature, and therefore can more accurately model the tire carcass without increasing the computation time. Thus, the 20-node isoparametric element is recommended for future study.
- The prediction of contact area on hard ground can be improved. Instead of using a rectangle to approximate the contact area, a more accurate contact area can be obtained using the method for viewing the normal pressure distribution for a given view plane, as shown in Figs. 5.26, 5.27 and 5.28. This means that if a view plane near the tire-hard ground interface is selected, the circumference of the contact area can then be obtained, and the contact area can be calculated.
- The method for deriving the ratio of equivalent Young's moduli α can be improved. The current method utilizes linear interpolation or extrapolation. It can be replaced by an incremental method.
- In the finite element tire-terrain interaction model developed, the influence of the change in contact area during slip-sinkage on the tire contact geometry is ignored. This assumption is made mainly to reduce computation time. If the change in contact geometry during slip-sinkage were taken into account, the computation time would be at least doubled. Therefore, if conditions permit in the future, this assumption should be eliminated from the model.
- If the two-equilibrium-position assumption is to be used for two dimensional analysis, it is beneficial to search for a better mathematical expression for the relationship between slip-sinkage and shear displacement. This is because the normal pressure predicted using the different expressions for the relationship between slip-sinkage and shear displacement are not the same, even though the

coefficients in these expressions are derived from the same experimental data.

- Terrain characteristics represented by the semi-empirical relationships obtained using the bevameter technique may be replaced using the finite element method to improve the accuracy of terrain representation. The possibility is shown by Nakashima et al [42]. In their soil-lug interaction model, the soil is modelled using the rigid-plastic finite element method incorporating two special techniques. One is the mesh rezoning technique for modeling large plastic deformation of the soil, and the other is the use of observation markers for tracing local soil deformation.
- The method for calculating shear displacement may be improved with further study on the kinematics of the tire-terrain system. In both the practical and the finite element tire-terrain interaction models, the influence of terrain movement under the action of the tire on the calculation of shear displacement is ignored. More realistic tire-terrain interaction model should take this into account.

•

References

- [1] Wong, J.Y., *Theory of Ground Vehicles*, John Wiley, New York, 1978.
- [2] Bekker, M.G., *Introduction to Terrain-Vehicle System*, University of Michigan, 1969.
- [3] Fujimoto, Y., *Performance of Elastic Wheel on Yielding Cohesive Soils*, Journal of Terramechanics, Vol.14, No.4, pp.191-210, 1977.
- [4] Wong, J.Y., and Preston-Thomas, J., *Development of Vehicle Performance Predicting Software*, Report Prepared for the National Research Council of Canada, under Contract Serial No. OS285-00054, March, 1986.
- [5] Baladi, G.Y., and Rohani, B., *Development of a Soil-wheel Interaction Model*, Proceedings of the Eighth International Conference of ISTVS, Cambridge, U.K., 1984.
- [6] Krick, G., *Radial and Shear Stress Distribution under Rigid Wheels and Pneumatic Tires Operating on Yielding Soils with Consideration of Tire Deformation*, Journal of Terramechanics, Vol.6, No.3, pp.73-98, 1969.
- [7] Oida, A., Satoh, A., Itoh, H., and Tiratanasirichia, K., *Three-dimensional Stress Distributions on A Tire-Sand Contact Surface*, Journal of Terramechanics, Vol.28, No.4, pp.319-330, 1991.
- [8] Wulfsohn, D., and Upadhyaya, S.K., *Prediction of Traction and Soil Compaction Using Three-Dimensional Soil-Tyre Contact Profile*, Journal of Terramechanics, Vol.29, No.6, pp.541-564, 1992.

- [9] Wulfsoln, D., and Upadhyaya, S.K., *Determination of Dynamic Three-dimensional Soil-tyre Contact Profile*, Journal of Terramechanics, Vol.29, No.4/5, pp.433-464, 1992.
- [10] Perumpral, J.V., Liljedahl, J.B., and Perloff, W.H., *A Numerical Method for Predicting the Stress Distribution and Soil Deformation under a Tractor Wheel*, Journal of Terramechanics, Vol.8, No.1, pp.9-20, 1971.
- [11] Yong, R.N., Fattah, E.A., and Bounsinsuk, P., *Analysis and Prediction of Tire-Soil Interaction and Performance Using Finite Elements*, Journal of Terramechanics, Vol.15, No.1, pp.43-63, 1978.
- [12] Oida, A., *Analysis of Rheological Deformation of Soil by Means of Finite Element Method*, Journal of Terramechanics, Vol.21, No.3, pp.237-251, 1984.
- [13] Chen, S.T., *Study of Tire-Terrain Interaction Using Finite Element Method*, M.Eng. Thesis, Department of Mechanical and Aeronautical Engineering, Carleton University, 1988.
- [14] Chang, T.Y., Saleeb, A.F., and Shyu, S.C., *Finite Element Solutions of Two-dimensional Contact Problems Based on a Consistent Mixed Formulation*, Computers and Structures, Vol.27, No.4, pp.445-466, 1987.
- [15] Oden, J.T., and Lin, T.L., *On the General Rolling Contact Problem for Finite Deformation of A Viscoelastic Cylinder*, Computer Method in Applied Mechanics and Engineering, Vol.57, pp.297-367, 1986.
- [16] Chen, W.H., and Tsai, P., *Finite Element Analysis of Elastodynamics Sliding Contact Problems with Friction*, Computers and Structures, Vol.22, No.6, pp.925-936, 1986.
- [17] Whicker, D., and Rohde, S.M., *Modelling Tire Deformation for Power Loss Calculations*, SAE Trans. 810161, 1981.

- [18] Segalman, D.J., *Modeling Tire Energy Dissipation for Power Loss Calculations*, SAE Trans. 810162, 1981.
- [19] Browne, A.L., and Arambages, A., *Modeling Tire Thermal State of Tires for Power Loss Calculations*, SAE Trans. 810163, 1981.
- [20] Ridha, R.A., Satyamurthy, K., and Hirschfeld, L.R., *Finite Element Modeling of A Homogeneous Pneumatic Tire Subjected to Footprint Loadings*, Tire Science and Technology, Vol.13, No.2, pp.91-110, April-June 1985.
- [21] Kenny, T.M., and Stechschulte, R.A., *Application of Finite Element Analysis in Tire Design*, Tire Science and Technology, TSTCA, Vol.16, No.2, pp.96-117, April-June, 1988.
- [22] Faria, L.O., Bass, J.M., Oden, J.T., and Bekker, E.B., *A Three-dimensional Rolling Contact Model for a Reinforced Rubber Tire*, Tire Science and Technology, TSTCA, Vol.17, No.3, July-September, pp.217-233, 1989.
- [23] Faria, L.O., Oden, J.T., Yavari, B., Tworzydlo, W.W., Barss, J.M., and Bekker, E.B., *Tire Modelling by Finite Elements*, Tire Science and Technology, TSTCA, Vol.20, No.1, January-March, pp.33-56, 1992.
- [24] Gall, R., Tkacik, P., and Andrews, M., *On the Incorporation of Frictional Effects in the Tire/Ground Contact Area*, Tire Science and Technology, TSTCA, Vol.21, No.1, January-March, pp.2-22, 1993.
- [25] Nakashima, H., and Wong, J.Y., *A Three-dimensional Tire Model by the Finite Element Method*, Journal of Terramechanics, Vol.30, No.1, pp.21-34, 1993.
- [26] Wang, L., *Tire Modelling Using a Three-dimensional Finite Element Method*, M.Eng. Thesis, Department of Mechanical and Aeronautical Engineering, Carleton University, 1990.

- [27] Chaudhary A.B., and Bathe, K.J., *A Solution Method for Static and Dynamic Analysis of Three-dimensional Contact Problems with Friction*, Computers and Structures, Vol.24, No.6, pp.855-873, 1986.
- [28] Bekker, M.G., *Off-the-Road Locomotion*, The university of Michigan Press, 1960.
- [29] Reece, A.R., *Principles of Soil-Vehicle Mechanics*, Proceedings of Institution of Mechanical Engineers, Vol.180, Part 2A, No.2, 1965-1966.
- [30] Bathe, K.J. *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, Inc., 1982.
- [31] Bhatt, G., Touzot, G., and Cantin, G., *The Finite Element Method Displayed*, John Wiley and Sons, 1984.
- [32] Wong, J.Y., Garber, M., Radforth, J.R., and Dowell, J.T., *Characterization of the Mechanical Properties of Muskeg with Special Reference to Vehicle Mobility*, Journal of Terramechanics, Vol.16, No.4, pp.163-180, 1979.
- [33] Wong, J.Y., and Preston-Thomas, J., *On the Characterization of the Pressure-Sinkage Relationship of Snow Covers Containing an Ice Layer*, Journal of Terramechanics, Vol.20, No.1, pp.1-12, 1983.
- [34] Wong, J.Y., Irwin, G.J., *Measurements and Characterization of the Pressure-Sinkage Data for Snow Obtained Using a Rammsonde*, Journal of Terramechanics, Vol.29, No.2, pp.265-280, 1992.
- [35] Wong, J.Y., Garber, M., and Preston-Thomas, J., *Theoretical Prediction and Experimental Substantiation of the Ground Pressure Distribution and Tractive Performance of Tracked Vehicles*, Proceedings of the Institution of Mechanical Engineers, Part D, No.4, 1984.
- [36] Wong, J.Y., *Computer-Aided Analysis of the Effects of Design Parameters on the Performance of Tracked Vehicles*, Journal of Terramechanics, Vol.23, No.2, pp.95-124, 1986.

- [37] Wong, J.Y., *Computer-Aided Methods for Optimization of the Mobility of Single-Unit and Two-Unit Articulated Tracked Vehicles*, Journal of Terramechanics, Vol.29, No.4/5, pp.359-412, 1992.
- [38] Bekker, M.G., *Theory of Load Locomotion*, University of Michigan, 1956.
- [39] Wong, J.Y., *Data Processing Methodology in the Characterization of the Mechanical Properties of Terrain*, Journal of Terramechanics, Vol.17, No.1, pp.13-41, 1980.
- [40] Liu, D.G., Fei, J.G., Yu, Y.J., and Li, G.Y., *Collected Numerical Algorithms in FORTRAN*, Vol.2, pp.129-141, 1980 (in Chinese).
- [41] Wong, J.Y., *Study of Tire-Terrain Interaction*, Journal of Terramechanics, Vol.18, No.3, pp.112-134, 1981.
- [42] Nakashima, H., Tanaka, T., and Yamazaki, M., *Finite Element Analysis of Soil-Lug Interactions*, Proceedings of 10th international Conference of the ISTVS, Kobe, Japan, August 20-24, 1990.
- [43] Marquardt, D.W., *An Algorithm for Least squares Estimation of Nonlinear Parameters*, JSIAM, Vol.11, No.2, pp.431-441, 1963.
- [44] Grahn, M., *Prediction of Sinkage and Rolling Resistance for Off-the-Road Vehicles Considering Penetration Velocity*, Journal of Terramechanics, Vol.28, No.4, pp.339-347, 1991.
- [45] Hinton, E., and Owen, D.R.J., *Finite Element Programming*, Academic Press, London, 1977.
- [46] Miller, C., Popper, P., Gilmour, P.W., and Schaffers, W.J., *Textile Mechanics Method of a Pneumatic Tire*, Tire Science and Technology, TSTCA, Vol.13, No.4, pp.187-226, October-December, 1985.
- [47] Nakajima, Y., and Padovan, J., *Numerical Simulation of Tire Sliding Events Involving Impacts with Holes and Bumps*, Tire Science and Technology, TSTCA, Vol.14, No.2, pp.125-136, April-June 1986.

- [48] Rothert, H., and Gall, R., *On the Three Dimensional Computation of Steel-Belted Tires*, Tire Science and Technology, TSTCA, Vol.14, No.2, pp.116-124, April-June 1986.
- [49] Sarkar, K., Kvyon, Y.D., and Prevorsek, D.C., *A New Approach for the Thermomechanical Analysis of Tires by Finite Element Method*, Tire Science and Technology, TSTCA, Vol.15, No.4, pp.161-275, October-December 1987.
- [50] Stechschute, R.A., and Luchini, J.R., *A Laminated Composite Solid Element and Its Application to Tire Analysis*, Tire Science and Technology, TSTCA, Vol.15, No.1, pp.42-57, 1987.
- [51] Timoshenko, S.P., and Goodier, J.N., *Theory of Elasticity*, McGraw-Hill Book Company, 3rd edition 1970.
- [52] Tseng, Nien-Tsu, *Finite Element Analysis of Freely Rolling Tires*, Tire Science and Technology, TSTCA, Vol.15, No.2, pp.134-158, April-June, 1987.
- [53] Upadhyaya, S.K., Chancellor, W.J., Wulfsohn, D., and Glancey, J.L., *Sources of Variability in Traction Data*, Journal of Terramechanics, Vol.25, No.4, pp.249-272, 1988.
- [54] Wong J.Y., *On the Study of Wheel-Soil Interaction*, Journal of Terramechanics, Vol.21, No.2, pp.117-131, 1984.
- [55] Yong, R.N., Boosinsuk, P., and Fattah, E.A., *Prediction of Tire Performance on Soils Related to Carcass Stiffness and Contact Areas*, Proceedings of the Sixth International Conference of ISTVS, Vol.II, Vienna, Austria, 1978.

Appendix A

Formulation of Least Squares Method for Fitting Nonlinear Expressions

To find the nonlinear parameter B in the function $y = f(x, B)$ for the given data set $(x_i, y_i), i = 1, 2, \dots, n$, the following function has to be minimized,

$$F = \sum_{K=1}^n [y_K - f(x_K, B)]^2 \quad (\text{A.1})$$

where x can be single variable or of P variables, that is,

$$x = (x_{1K}, x_{2K}, \dots, x_{PK}) \quad K = 1, 2, \dots, n \quad (\text{A.2})$$

The parameter B can be single or of m elements, that is,

$$B = (b_1, b_2, \dots, b_m) \quad (m \leq n) \quad (\text{A.3})$$

To solve such problems involving nonlinear parameters, a set of initial values of b_i , noted as $b_i^{(0)}$, is provided, and the difference between the initial values and the true values is defined as Δ_i , thus,

$$b_i = b_i^{(0)} + \Delta_i \quad i = 1, 2, \dots, m \quad (\text{A.4})$$

Thus, finding b_i is equivalent to finding Δ_i . To find b_i , expanding f in Taylor series at $b_i^{(0)}$ and omitting the higher order terms, one obtains,

$$f(x_K, b_1, b_2, \dots, b_m) = f_{K0} + \frac{\partial f_{K0}}{\partial b_1} \Delta_1 + \frac{\partial f_{K0}}{\partial b_2} \Delta_2 + \dots + \frac{\partial f_{K0}}{\partial b_m} \Delta_m \quad (\text{A.5})$$

where

$$f_{K0} = f(x_K, b_1^{(0)}, b_2^{(0)}, \dots, b_m^{(0)}) \quad (\text{A.6})$$

and

$$\frac{\partial f_{K0}}{\partial b_i} = \frac{\partial f(x, b_1, b_2, \dots, b_m)}{\partial b_i} \quad (\text{A.7})$$

when $x = x_K$ and $b_i = b_i^{(0)}$.

From the principle of least squares, one can obtain,

$$\left. \begin{aligned} F &= \sum_{K=1}^n [y_K - f(x_K, b_1, b_2, \dots, b_m)]^2 \\ &= \sum_{K=1}^n \left[y_K - \left(f_{K0} + \frac{\partial f_{K0}}{\partial b_1} \Delta_1 + \frac{\partial f_{K0}}{\partial b_2} \Delta_2 + \dots + \frac{\partial f_{K0}}{\partial b_m} \Delta_m \right) \right]^2 \end{aligned} \right\} \quad (\text{A.8})$$

and

$$\left. \begin{aligned} \frac{\partial F}{\partial \Delta_i} &= \frac{\partial F}{\partial \Delta_i} \\ &= 2 \left[\Delta_i \sum_{K=1}^n \frac{\partial f_{K0}}{\partial b_1} \frac{\partial f_{K0}}{\partial b_i} + \dots \right. \\ &\quad \left. + \Delta_m \sum_{K=1}^n \frac{\partial f_{K0}}{\partial b_m} \frac{\partial f_{K0}}{\partial b_i} - \sum_{K=1}^n \frac{\partial f_{K0}}{\partial b_i} (y_K - f_{K0}) \right] \end{aligned} \right\} \quad (\text{A.9})$$

Note

$$\left. \begin{aligned} a_{ij} &= \sum_{K=1}^n \frac{\partial f_{K0}}{\partial b_i} \frac{\partial f_{K0}}{\partial b_j} \quad i, j = 1, 2, \dots, m \\ a_i &= \sum_{K=1}^n \frac{\partial f_{K0}}{\partial b_i} (y_K - f_{K0}) \quad i = 1, 2, \dots, m \end{aligned} \right\} \quad (\text{A.10})$$

Therefore, m simultaneous linear equations of $\Delta_i (i = 1, 2, \dots, m)$ can be obtained,

$$\left. \begin{aligned} a_{11}\Delta_1 + a_{12}\Delta_2 + \dots + a_{1m}\Delta_m &= a_{1y} \\ a_{21}\Delta_1 + a_{22}\Delta_2 + \dots + a_{2m}\Delta_m &= a_{2y} \\ &\dots \\ a_{m1}\Delta_1 + a_{m2}\Delta_2 + \dots + a_{mm}\Delta_m &= a_{my} \end{aligned} \right\} \quad (\text{A.11})$$

The coefficients a_{ij} and $a_i y$ on the right-hand side of Eq. A.11 can be computed for a given set of data (x_K, y_K) , $K = 1, 2, \dots, n$ and initial values of $b_i^{(0)}$, $i = 1, 2, \dots, m$. Therefore, from Eq. A.11 Δ_i can be solved. Then the values of b_i can be obtained using Eq. A.4. If $|\Delta_i|$ is large, the $b_i^{(0)}$ is replaced with current b_i , and a_{ij} and $a_i y$ are re-computed, and a new set of Δ_i can be obtained. The process is repeated until $|\Delta_i|$ is sufficiently small.

Appendix B

Computer Source Codes for Simulation of Tire-Hard Ground Interaction — Finite Element Model

Program TRACG is designed to generate finite element meshes for tire models

Program TRAC1 is designed to calculate two sets of contact area deflection relationships for two given values of the ratio of Young's moduli α .

Program TRACM11 is designed to process the output files from TRAC1 to obtain the relationship between the ratio of Young's moduli α and tire deflection.

Program TRAC2 is designed to obtain the equivalent Young's moduli for the tire tread, E_t , and for the side wall of the tire, E_s .

Program TRACM2 is designed to process the output files from TRAC2 to obtain the polynomial coefficients for the relationships between Young's moduli and normal load.

Program TRAC3 is designed to verify the finite element tire model.

```

C .....
C CALL BCSET(IPROB,NDIV,NCIRC)
C .....
C REMEMBERING OF NODE NUMBERS (Currently Not Effective)
C .....
C BEGIN=1
C CALL REMBER
C IF(RENUMEQ.1) CALL DCONV
C .....
C GENERATED DATA SAVE [gene3d.dat]
C .....
C CALL DSAVE(ITITLE,PRESS,FRCFAC,NM,NMAT)
C STOP
C END
C .....
C SUBROUTINE DATIN(ITITLE,IPROB,RW,PRESS,FRCFAC,NM)
C
C This subroutine reads input data, the cross-section
C geometry of the tire carcass and material properties.
C
C IMPLICIT REAL*8 (A-H,O-Z)
C CHARACTER*40 ITITLE
C PARAMETER(NIRGN=6,NMAX=30,NMAXX=4)
C COMMON /SECT/ ZDATA(2,NMAX),YDATA(2,NMAX),ZSECT(2,6,3),
C YSECT(2,6,3)
C COMMON /MATPR/ YOUNG(NMAX),POISS(NMAX),NMIR(NMAX,2)
C COMMON /SECT1/ NSECT(NMAX),NDIVS(NRGN),NCIRC(3),NPNT,NDIV
C DIMENSION ZSECT(2,30),YSECT(2,30)
C
C OPEN(5,FILE='section.dat')
C WRITE(';',*) '--- Welcome to T R A C G ---'
C WRITE(';',*) ' > DATIN'
C READ(5,'(A40)') ITITLE
C READ(5,'') RW,PRESS,FRCFAC,IPROB
C READ(5,'') NPNT
C
C DO 10 I=1,2
C DO 20 J=1,NPNT
C READ(5,'') ZSECT(I,J),YSECT(I,J),NSECT(I,J)
C 20 CONTINUE
C 10 CONTINUE
C
C DO 30 I=1,NSECT(NPNT)
C READ(5,'') NDIVS(I)
C 30 CONTINUE
C
C DO 40 I=1,3
C READ(5,'') NCIRC(I)
C 40 CONTINUE
C
C DO 50 I=1,2

```

```

C .....
C TRACG.f90
C -- Data Generator for TRAC 3D FEM Analysis --
C
C Hiroshi Hatakeyama
C Transport Technology Research Laboratory
C Carlsson University
C
C Input Data File --> 'section.dat'
C Output Data File(1) --> 'secpit.dat'
C Output Data File(2) --> 'gene3d.dat'
C
C (Ver. 1.5, June 12, 1991)
C .....
C IMPLICIT REAL*8 (A-H,O-Z)
C CHARACTER*40 ITITLE
C PARAMETER(NMAX=30,NRGN=6)
C PARAMETER(NPMAX=800,NEMAX=500,MAXCON=28,NMAXX=4)
C COMMON /COORD/ X(NPMAX),Y(NPMAX),Z(NPMAX)
C COMMON /ELNOD/ LNODE(NEMAX,8)
C COMMON /BOUND/ NBC(NPMAX,3)
C COMMON /BINEF/ NPE,DIFF,IRENUM,IBEGIN
C COMMON /MATPR/ YOUNG(NMAX),POISS(NMAX),NMIR(NMAX,2)
C COMMON /NUMBERING/INT(NPMAX),MEMJT(NPMAX*MAXCON),MEM(NPMAX)
C COMMON /SECT/ ZDATA(2,NMAX),YDATA(2,NMAX),ZSECT(2,6,3),
C YSECT(2,6,3)
C COMMON /SECT1/ NSECT(NMAX),NDIVS(NRGN),NCIRC(3),NPNT,NDIV
C DIMENSION S,HAPE(3),GZAR(10)
C DIMENSION NMAT(NEMAX)
C
C ORIGINAL DATA INPUT [section.dat]
C
C CALL DATIN(ITITLE,IPROB,RW,PRESS,FRCFAC,NM)
C
C SECTIONAL DATA GENERATION (Z-axis and Y-axis plane)
C
C CALL DGSECT
C
C CIRCUMFERENTIAL DATA GENERATION (Rotation w.r.t. Z-axis)
C
C CALL DGCIRC(RW,IPROB)
C
C ELEMENT-NODE RELATION SETUP
C
C CALL ELNODD(IPROB,NMAT,NM)
C
C SECTIONAL DATA SAVE [secpit.dat]
C
C CALL SECPIT(ITITLE,NDIV)
C
C BOUNDARY CONDITION SETUP

```

```

DO 60 J=1,NPNT,2
  NSECT1=NSECT(J)
  ZSECT1((NSECT1,1)=ZSECT((J,J)
  YSECT1((NSECT1,1)=YSECT((J,J)
  IF (J=1.EQ.NPNT) THEN
    ZSECT1((NSECT1,2)=ZSECT((J,J+1)
    YSECT1((NSECT1,2)=YSECT((J,J+1)
    GOTO 50
  ELSE
    ZSECT1((NSECT1,2)=ZSECT((J,J+1)
    YSECT1((NSECT1,2)=YSECT((J,J+1)
    ZSECT1((NSECT1,3)=ZSECT((J,J+2)
    YSECT1((NSECT1,3)=YSECT((J,J+2)
  ENDF
60 CONTINUE
50 CONTINUE
C
  READ(S,'') NM
  DO 70 I=1,NM
    READ(S,'') NMRR(I,1),NMRR(I,2),YOUNG(I),POISS(I)
  70 CONTINUE
  CLOSE(5)
C
  RETURN
END
C
SUBROUTINE DGSECT
C
C This subroutine generates some additional data points by
C interpolating the input data points
C
IMPLICIT REAL*8 (A-H,O-Z)
PARAMETER(NDMAX=30,NRGN=6)
COMMON /SECT/ ZDATA(2,NDMAX),YDATA(2,NDMAX),ZSECT1(2,6,3),
  YSECT1(2,6,3)
COMMON /SECT1/ NSECT(NDMAX),NDIVS(NRGN),NCIRC(3),NPNT,NDIV
DIMENSION SHAPE(3),GZAX(16),
WRITE('') ' > DGSECT'
DO 10 I=1,2
  L=0
  DO 20 J=1,NSECT(NPNT)
    GZAX(I)=1.000
    DO 30 K=2,NDIVS(J)
      GZAX(K)=GZAX(K-1)+2.000D0/LOAT(NDIVS(J))
    CONTINUE
    DO 40 K1=1,NDIVS(J)
      L=L+1
      IF (J.EQ.NSECT(NPNT)) THEN
        ZDATA(I,L)=ZSECT1(I,J,1)
        YDATA(I,L)=YSECT1(I,J,1)
        ZDATA(I,L+1)=ZSECT1(I,J,2)
        YDATA(I,L+1)=YSECT1(I,J,2)
      ELSE
        YDATA(I,L+1)=YSECT1(I,J,2)
      ELSE
        CALL FUNCS(GZAX(K1),SHAPE)
        ZSUM=0.000
        YSUM=0.000
        DO 50 K2=1,3
          ZSUM=ZSUM+ZSECT1(I,J,K2)*SHAPE(K2)
          YSUM=YSUM+YSECT1(I,J,K2)*SHAPE(K2)
        CONTINUE
        ZDATA(I,L)=ZSUM
        YDATA(I,L)=YSUM
      ENDF
    40 CONTINUE
    NDIV=L+1
  10 CONTINUE
  RETURN
END
C
SUBROUTINE FUNCS(GZAX,SHAPE)
C
C This subroutine provides a set of quadratic shape functions for
C interpolation.
C
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION SHAPE(3)
SHAPE(1)=-0.5*(1.0-GZAX)*GZAX
SHAPE(2)= 1.0-GZAX*GZAX
SHAPE(3)= 0.5*(1.0+GZAX)*GZAX
RETURN
END
C
SUBROUTINE DGCIRC(RW,IPROB)
C
C This subroutine divides a half tire model or a whole tire model
C into several sections so that the part in contact with ground
C has finer mesh
C
IMPLICIT REAL*8 (A-H,O-Z)
PARAMETER(NDMAX=30,NRGN=6)
PARAMETER(NPMAX=800,NEMAX=500)
COMMON /COORD/ X(NPMAX),Y(NPMAX),Z(NPMAX)
COMMON /MEINF/ NP,NF,DIFF,IREN,JM,BEGIN
COMMON /SECT/ ZDATA(2,NDMAX),YDATA(2,NDMAX),ZSECT1(2,6,3),
  YSECT1(2,6,3)
COMMON /SECT1/ NSECT(NDMAX),NDIVS(NRGN),NCIRC(3),NPNT,NDIV
D2R=3.141592653589793D0/180.D0
WRITE('') ' > DGCIRC'
C
C .. CONVERSION OF ORIGIN OF Y-AXIS TO TIRE ROTATION CENTER

```



```
C
C      LM(K)=L
DO 120 I=1,L
WRITE(6,' ')I,X1(I,K),Y1(I,K),Y2(I,K)
120 CONTINUE
C
C      5 CONTINUE
RETURN
END
C-----
SUBROUTINE VSETS(I,J)
C
C      This subroutine excludes zeros from Y1 (contact width)
C      and Y2 (contact length), and saves either of them in RX
C      XP is deflection
C
C      IMPLICIT REAL*8 (A-H,O-Z)
COMMON /PNTDD/ XP(50),RX(50),NP
COMMON /DATAF/ X1(80,2),Y1(80,2),Y2(80,2),LM(2)
C
C      L1=0
DO 10 K=1,LM(J)
IF ((IEQ 1) AND (Y1(K,J) NE 0.0D0)) THEN
C
C      ... contact width
L1=L1+1
RX(L1)=Y1(K,J)
ELSEIF ((IEQ 2) AND (Y2(K,J) NE 0.0D0)) THEN
C
C      ... contact length
L1=L1+1
RX(L1)=Y2(K,J)
ENDIF
XP(L1)=X1(K,J)
10 CONTINUE
NP=L1
RETURN
END
C-----
SUBROUTINE REGS(IMODE)
C
C      This subroutine obtains the regression coefficients
C
C      IMPLICIT REAL*8 (A-H,O-Z)
COMMON /PARAG/ AR(10),NAR
COMMON /PNTDD/ XP(50),RX(50),NP
REAL*8 A(50,25),AWA(25,25),W(50)
C
C      IF (IMODE.EQ.0) THEN
C      NAR=N3
C      ELSE
WRITE(6,' ') '> Order of Regression Equation ?'

```

```

C      & POSITIVE Y-Axis DIRECTION IS SET TO UPWARD --
C
DO 80 I=1,2
DO 80 J=1,NDIV
YDATA(I,J)=RW-YDATA(I,J)
80 CONTINUE
C
K=0
DO 85 L=1,3
DO 86 L1=1,NCIRC(L)
IF (L EQ 1) THEN
ANGFIX=45.
ANGORG=0.
ELSEIF (L EQ 2) THEN
ANGFIX=45.
ANGORG=45.
ELSEIF (L EQ 3) THEN
ANGFIX=90.
ANGORG=90.
ENDIF
ANG=(ANGORG+ANGFIX/FLOAT(NCIRC(L))*FLOAT(L-1))*D2R
DO 90 I=1,2
DO 100 J=1,NDIV
XO=0.
YO=YDATA(I,J)
ZO=ZDATA(I,J)
CALL ROTATE(XN,YN,ZN,XO,YO,ZO,ANG,3)
K=K+1
X(K)=XN
Y(K)=YN
Z(K)=ZN
100 CONTINUE
90 CONTINUE
85 CONTINUE
IF (IPROB EQ 0) GOTO 400
C      .. Rotate 'jn' from 180 deg to 360 deg for full tire model (IPROB=1) --
C
DO 87 L=3,1,-1
C
DO 88 L1=1,NCIRC(L)
IF (L EQ 3) THEN
ANGORG=180
ANGFIX=90
ELSEIF (L EQ 2) THEN
ANGORG=45
ANGFIX=45
ANGORG=270
ELSEIF (L EQ 1) THEN
ANGORG=45
ANGFIX=45
ANGORG=315
ENDIF
C
ANG=(ANGORG+ANGFIX/FLOAT(NCIRC(L))*FLOAT(L-1))*D2R
DO 95 I=1,2
DO 96 J=1,NDIV
XO=0.
YO=YDATA(I,J)
ZO=ZDATA(I,J)
CALL ROTATE(XN,YN,ZN,XO,YO,ZO,ANG,3)
K=K+1
X(K)=XN
Y(K)=YN
Z(K)=ZN
96 CONTINUE
95 CONTINUE
87 CONTINUE
C
C      .. Final node set up for half tire model (IPROB=0) --
C
400 IF (IPROB EQ 0) THEN
ANG=180.0*D2R
DO 110 I=1,2
DO 110 J=1,NDIV
XO=0.
YO=YDATA(I,J)
ZO=ZDATA(I,J)
CALL ROTATE(XN,YN,ZN,XO,YO,ZO,ANG,3)
K=K+1
X(K)=XN
Y(K)=YN
Z(K)=ZN
110 CONTINUE
ENDIF
C
NP=K
RETURN
END
C-----
SUBROUTINE ROTATE(XN,YN,ZN,XO,YO,ZO,ANG,IMODE)
C
C      This subroutine performs the coordinate rotation
C
C
IMPLICIT REAL*8 (A-H,O-Z)
IMODE=1 X-AXIS ROTATION, IMODE=2 Y-AXIS ROTATION,
IMODE=3 Z-AXIS ROTATION
IF (IMODE EQ 1) THEN
XN=XO
YN=YO*cos(ANG)+ZO*sin(ANG)
ZN=YO*sin(ANG)+ZO*cos(ANG)
RETURN
ELSEIF (IMODE EQ 2) THEN
XN=XO*cos(ANG)+ZO*sin(ANG)
YN=YO
ZN=YO*sin(ANG)+ZO*cos(ANG)
ENDIF
END

```

```

      ZN=XO*SIN(ANG)+ZO*COS(ANG)
      RETURN
    ELSEIF (IMODE.EQ.3) THEN
      XN=XO*COS(ANG)+YO*SIN(ANG)
      YN=XO*SIN(ANG)+YO*COS(ANG)
      ZN=ZO
      RETURN
    ENDIF
    RETURN
  END
END
C-----
SUBROUTINE ELMNOD(IPROB,NMAT,NM)
C
C This subroutine performs the node numbering for the finite
C element mesh
C
  IMPLICIT REAL*8 (A-H,O-Z)
  CHARACTER*40 ITITLE
  PARAMETER(NOMAX=30,NRGN=6,NMMAX=4)
  PARAMETER(NPMAX=800,NEMAX=500)
  COMMON /COORD/ X(NPMAX),Y(NPMAX),Z(NPMAX)
  COMMON /ELNOD/ LNODE(NEMAX,8)
  COMMON /MEINF/ NP,NE,DIFF,IRENUM,IBEGIN
  COMMON /SECT/ ZDATA(2,NOMAX),YDATA(2,NOMAX),ZSECT1(2,6,3),
    * YSECT1(2,6,3)
  COMMON /SECT1/ NSECT(NOMAX),NDIVS(NRGN),NCIRC(3),NPNT,NDIV
  COMMON /MATPR/ YOUNG(NMMAX),POISS(NMMAX),NMER(NMMAX,2)
  DIMENSION NMAT(NEMAX)
C
  WRITE(' ') > ELMNOD
  NEDIV=NDIV-1
  NE=NCIRC(1)+NCIRC(2)+NCIRC(3)
  L=0
  DO 2 K=1,NSECT(NPNT)
    L=L+NDIVS(K)
  2 CONTINUE
  NE=NE+L
  IF (IPROB.EQ.1) NE=NE*2
C
C -- Region 1 for tread can have different rigidity --
C
  IE=0
  ISECT=0
  IELAST=NE-NEDIV
  500 IE=IE+1
  K=INT((IE-1)/NEDIV)
  IF (ISECT.EQ.NEDIV) ISECT=0
  ISECT=ISECT+1
  DO 10 IM=1,NM
    NMLAST=NMR(IM,2)
    NRLAST=0
    DO 15 NI=1,NMLAST
      NRLAST=NRLAST+NDIVS(NI)
    15 CONTINUE
    NMRGN=NMR(IM,1)
    NRBGN=0
    DO 16 NI=2,NMRGN
      NRBGN=NRBGN+NDIVS(NI-1)
    16 CONTINUE
    WRITE(' ') IE,ISECT,NRBGN,NRLAST
    IF ((ISECT.GT.NRBGN) .AND. (ISECT.LE.NRLAST)) THEN
      NMAT(IE)=IM
      GOTO 770
    ENDIF
  10 CONTINUE
C
  770 CONTINUE
  LNODE(IE,1)=2*NDIV*K +ISECT
  LNODE(IE,2)=2*NDIV*K +ISECT+1
  IF ((IPROB.EQ.1) .AND. (IE.GT.IELAST)) THEN
    LNODE(IE,3)=ISECT+1
    LNODE(IE,4)=ISECT
  ELSE
    LNODE(IE,3)=2*NDIV*(K+1) +ISECT+1
    LNODE(IE,4)=2*NDIV*(K+1) +ISECT
  ENDIF
  LNODE(IE,5)=2*NDIV*K+NDIV +ISECT
  LNODE(IE,6)=2*NDIV*K+NDIV +ISECT+1
  IF ((IPROB.EQ.1) .AND. (IE.GT.IELAST)) THEN
    LNODE(IE,7)=NDIV+ISECT+1
    LNODE(IE,8)=NDIV+ISECT
  ELSE
    LNODE(IE,7)=2*NDIV*(K+1)+NDIV+ISECT+1
    LNODE(IE,8)=2*NDIV*(K+1)+NDIV+ISECT
  ENDIF
  IF (IE.EQ.NE) GOTO 600
  GOTO 500
600 CONTINUE
  RETURN
  END
C-----
SUBROUTINE RENMBR
C
C This subroutine has not yet been used or tested.
C
C----- MESH RENUMBERING BY COLLINS ALGORITHM(A) -----
C
  IMPLICIT REAL*8 (A-H,O-Z)
  PARAMETER(NPMAX=800,NEMAX=500,MAXCON=26)
  COMMON /ELNOD/ LNODE(NEMAX,8)
  COMMON /MEINF/ NP,NE,DIFF,IRENUM,IBEGIN
  COMMON /NUMBRNG/JNT(NPMAX),MEMJT(NPMAX*MAXCON),MEM(NPMAX)
  EQUIVALENCE (LNODE(1,1),JT(1))
  DIMENSION JT(NEMAX*8)
C

```

```

WRITE(';',) > RENMBR'
WRITE(';',) Node,Elem=,np,ne
C
IBEGIN=1
IDIFF=0
NCN=8
DO 10 I=1,NPMAX
  JNT(I)=0
  JMEM(I)=0
10 CONTINUE
IJ=NPMAX*MAXCON
DO 20 I=1,IJ
  MEMUT(I)=0
20 CONTINUE
DO 70 J=1,NE
  DO 60 I=1,NCN
    JNTI=JNT(NEMAX*(I-1)+J)
    IF (JNTI.EQ.0) GOTO 70
    JSUB=(JNTI-1)*MAXCON
    DO 50 II=1,NCN
      IF (II.EQ.I) GOTO 50
      JUT=JNT(NEMAX*(II-1)+J)
      IF (JUT.EQ.0) GOTO 60
      MEMI=JMEM(JNTI)
      IF (MEMI.EQ.0) GOTO 40
      DO 30 III=1,MEMI
        IF (MEMUT(JSUB+III).EQ.JUT) GOTO 50
        CONTINUE
      JMEM(JNTI)=JMEM(JNTI)+1
      MEMUT(JSUB+JMEM(JNTI))=JUT
      IF (IABS(JNTI-JUT) GT IDIFF) IDIFF=IABS(JNTI-JUT)
50 CONTINUE
60 CONTINUE
70 CONTINUE
WRITE(';',) >> Original node difference = ,IDIFF
CALL OPTNUM
RETURN
END
C-----
SUBROUTINE OPTNUM
C
C This subroutine has not yet been used or tested
C
C----- MESH RENUMBERING BY COLLINS ALGORITHM (B) -----
C
IMPLICIT REAL*8 (A-H,O-Z)
PARAMETER(NPMAX=800, NEMAX=500, MAXCON=26)
COMMON /ELNOD, LNODE(NEMAX,8)
COMMON /MEMINF, NP,NE IDIFF,IRENUM,IBEGIN
COMMON /NUMBRNG,JNT(NPMAX),MEMUT(NPMAX*MAXCON),JMEM(NPMAX)
EQUIVALENCE (LNODE(1,1),JT(1))
DIMENSION JT(NEMAX*8),JOINT(NPMAX),NEWJUT(NPMAX)
C

```

```

WRITE(';',) > OPTNUM'
IRENUM=1
MINMAX=IDIFF
C
DO 70 IK=1,NP
  DO 10 J=1,NP
    JOINT(J)=0
    NEWJUT(J)=0
10 CONTINUE
  MAX=0
  I=1
  NEWJUT(I)=IK
  JOINT(IK)=1
  K=1
20 CONTINUE
  K4=JMEM(NEWJUT(I))
  IF (K4.EQ.0) GOTO 40
  JSUB=(NEWJUT(I)-1)*MAXCON
  DO 30 JJ=1,K4
    K5=MEMUT(JSUB+JJ)
    IF (JOINT(K5) GT 0) GOTO 30
    K=K+1
    NEWJUT(K)=K5
    JOINT(K5)=K
    NDIFF=IABS(I-K)
    IF (NDIFF GE MINMAX) GOTO 70
    IF (NDIFF GT MAX ) MAX=NDIFF
30 CONTINUE
  IF (K EQ NP) GOTO 50
  I=I+1
  GOTO 20
50 MINMAX=MAX
  DO 60 J=1,NP
    JNT(J)=JOINT(J)
60 CONTINUE
70 CONTINUE
  IF (MINMAX LT IDIFF) GOTO 90
  IRENUM=0
  WRITE(';',) Original numbering scheme retained'
  WRITE(';',) Semi-bandwidth remains the same'
  GOTO 150
90 IDIFF=MINMAX
  WRITE(';',) >> Node difference after renumbering = ,IDIFF
150 CONTINUE
RETURN
END
C-----
SUBROUTINE BCSET(IPROB,NDIV,NCIRC)
C
C This subroutine sets the boundary conditions for each node
C
IMPLICIT REAL*8 (A-H,O-Z)

```

```

PARAMETER(NPMAX=800,NEMAX=500)
COMMON /BOUND/ NBC(NPMAX,3)
COMMON /MEINF/ NP,NE,DIFF,IRENUM,IBEGIN
DIMENSION NCIRC(3)
C
C  WRITE(' ',' > BCSET'
C
C  NBL=NCIRC(1)+NCIRC(2)+NCIRC(3)+1
C  -- NODE FIX : NBC(' ',1)=1; NODE FREE : NBC(' ',2)=0
C
C  DO 100 J=1,3
C  DO 100 I=1,NP
C  NBC(I,J)=0
100 CONTINUE
C  NDIV2=2*NDIV
C
C  DO 110 I=1,NBL
C  N11=NDIV2*(I-1)+NDIV-1
C  N12=NDIV2*(I-1)+NDIV
C  N24=NDIV2*(I-1)+NDIV2
C  N1 =NDIV2*(I-1)+1
C  N13=NDIV2*(I-1)+NDIV+1
C  DO 120 K=1,3
C  NBC(N11,K)=1
C  NBC(N12,K)=1
C  NBC(N24,K)=1
120 CONTINUE
C  NBC(N1 ,3)=1
C  NBC(N13,3)=1
C  IF (((IPROB.EQ.0) .AND. (I.EQ.1)) OR.
C  & ((IPROB.EQ.0) .AND. (I.EQ.NBL))) THEN
C  DO 112 J=1,NDIV
C  J1=NDIV2*(I-1)+J
C  J2=NDIV2*(I-1)+NDIV+J
C  NBC(J ,1)=1
C  NBC(J2,1)=1
112 CONTINUE
C  ENDF
110 CONTINUE
C
C  IF (IPROB.EQ.0) RETURN
C
C  DO 130 I=NBL+1,NBL*2-1
C  N11=NDIV2*(I-1)+NDIV-1
C  N12=NDIV2*(I-1)+NDIV
C  N24=NDIV2*(I-1)+NDIV2
C  N1 =NDIV2*(I-1)+1
C  N13=NDIV2*(I-1)+NDIV+1
C  DO 140 K=1,3
C  NBC(N11,K)=1
C  NBC(N12,K)=1
C  NBC(N24,K)=1

```

```

140 CONTINUE
NBC(N1 ,3)=1
NBC(N13,3)=1
130 CONTINUE
RETURN
END
C-----
SUBROUTINE DCONV
C
C  This subroutine has not yet been used or tested.
C
C  IMPLICIT REAL*8 (A-H,O-Z)
C  PARAMETER(NPMAX=800,NEMAX=500,MAXCON=26)
C  COMMON /COORD/ X(NPMAX),Y(NPMAX),Z(NPMAX)
C  COMMON /ELNOD/ LNODE(NEMAX,8)
C  COMMON /BOUND/ NBC(NPMAX,3)
C  COMMON /MEINF/ NP,NE,DIFF,IRENUM,IBEGIN
C  COMMON /NUMBRNG/JNT(NPMAX),MEMAJT(NPMAX*MAXCON),JMEM(NPMAX)
C
C  DIMENSION XBUF(NPMAX),YBUF(NPMAX),ZBUF(NPMAX)
C  DIMENSION LNBUFF(NEMAX,8),NBCBUF(NPMAX,3)
C
C  WRITE(' ',' > DCONV'
C  IBEGIN=JNT(1)
C  DO 10 I=1,NP
C  XBUF(I)=X(JNT(I))
C  YBUF(I)=Y(JNT(I))
C  ZBUF(I)=Z(JNT(I))
C  DO 20 K=1,3
C  NBCBUF(I,K)=NBC(JNT(I),K)
20 CONTINUE
10 CONTINUE
C  DO 30 I=1,NE
C  DO 30 J=1,8
C  LNBUFF(I,J)=JNT(LNODE(I,J))
30 CONTINUE
C
C  DO 50 I=1,NP
C  X(I)=XBUF(I)
C  Y(I)=YBUF(I)
C  Z(I)=ZBUF(I)
C  DO 60 K=1,3
C  NBC(I,K)=NBCBUF(I,K)
60 CONTINUE
50 CONTINUE
C  DO 70 I=1,NE
C  DO 70 K=1,8
C  LNODE(I,K)=LNBUFF(I,K)
70 CONTINUE
RETURN
END
C-----

```

```

C
C
C      This subroutine saves the generated finite element mesh
C      for the cross-section of the tire in file SECTPLT.DAT .
C
SUBROUTINE SEOPLT(ITTITLE,NDIV)
  IMPLICIT REAL*8 (A-H,O-Z)
  CHARACTER*40 ITITLE
  PARAMETER(NPMAX=800,NEMAX=500)
  COMMON /COORD/ X(NPMAX),Y(NPMAX),Z(NPMAX)
  COMMON /ELNOD/ LNODE(NEMAX,8)

  WRITE(' ') ' > SECTPLT'
  OPEN(10,FILE='sectplt.dat')
  WRITE(10, '(A40)') ITITLE
  WRITE(10, '2*NDIV,NDIV-1')
  DO 10 I=1,2*NDIV
    WRITE(10,930) Z(I),Y(I)
  10 CONTINUE
  930 FORMAT(2F15.5)
  DO 20 I=1,NDIV-1
    WRITE(10,940) LNODE(I,1),LNODE(I,2),LNODE(I,6),LNODE(I,5)
  20 CONTINUE
  940 FORMAT(4I5)
  CLOSE(10)
  WRITE(' ') ' . . . sectplt dat is saved '
  RETURN
END

C-----
C
C      SUBROUTINE DSAVE(ITTITLE,PRESS,FRCFAC,NM,NMAT)
C
C      This subroutine saves the generated finite element mesh
C      for the tire in file GENE3D.DAT .
C
  IMPLICIT REAL*8 (A-H,O-Z)
  CHARACTER*40 ITITLE
  PARAMETER(NPMAX=800,NEMAX=500,NMMAX=4)
  COMMON /COORD/ X(NPMAX),Y(NPMAX),Z(NPMAX)
  COMMON /ELNOD/ LNODE(NEMAX,8)
  COMMON /BOUND/ NBC(NPMAX,3)
  COMMON /MATPR/ YOUNG(NMMAX),POISS(NMMAX),NMIR(NMMAX,2)
  COMMON /MEINF/ NP,NE,DIFF,IRENUM,BEGIN
  DIMENSION NMAT(NEMAX)

  WRITE(' ') ' > DSAVE '
  OPEN(9,FILE='gene3d.dat')
  WRITE(9, '(A40)') ITITLE
  WRITE(9, '3') PRESS,FRCFAC
  WRITE(9,905) NP,NE,BEGIN
  905 FORMAT(3I5)
  DO 200 I=1,NP
    WRITE(9,900) I,X(I),Y(I),Z(I),NBC(I,K),K=1,3)
  200 CONTINUE

```

```

900 FORMAT(13,3F15.7,3I4)
DO 220 I=1,NE
  WRITE(9,910) I,(LNODE(I,K),K=1,8),NMAT(I)
910 FORMAT(13,4X,9I5)
220 CONTINUE
  WRITE(9, '3') NM
DO 300 I=1,NM
  WRITE(9,920) YOUNG(I),POISS(I)
300 CONTINUE
920 FORMAT(2E15.7)
CLOSE(9)
WRITE(' ') ' . . . gene3d dat is saved '
RETURN
END

```

```

C-----
C      KR=0
C      720 KR=KR+1
C      WRITE(6,971) KR
C      971 FORMAT(' ', : TRAC: Case Number', I2, ' ')
C-----
C      Parameter Setup
C-----
C      ISTAGE=0
C      TDEFL =0.0D+0
C      CALL NULVEC(FX,NPMAX)
C      CALL NULVEC(FY,NPMAX)
C      CALL NULVEC(FZ,NPMAX)
C-----
C      Data Input
C-----
C      WRITE(6,*) '*** T R A C / Step 1 in Process ***'
C      WRITE(6,*) 'DATAIN'
C      CALL DATIN(TITLE,PRESS,FRCFAC,IP1,DEFMAX)
C      WRITE(6,*) 'DEFMAX=',DEFMAX
C      CALL SETBC
C-----
C      Calculation of KDIAG
C-----
C      DO 10 I=1,NB
C      10 KDIAG(I)=0
C      DO 15 IE=1,NE
C      CALL IGSET(IE,IG)
C      15 CALL FKDIAG(KDIAG,IG,IDOF,IE,NEMAX)
C      KDIAG(1)=1
C      DO 20 I=2,NB
C      20 KDIAG(I)=KDIAG(I-1)+KDIAG(I-1)
C      NN=KDIAG(NB)
C      IF (NN.GT. IKV) CALL EMSSGE(1,NN,IKV)
C-----
C      Loop Entry of Incremental Contact
C-----
C      500 CONTINUE
C      ISTAGE=ISTAGE+1
C-----
C      Loop Entry of Penetration & Param Search
C-----
C      600 CONTINUE
C      CALL NULVEC(TREAC,NB)
C      IF (ISTAGE.EQ.1) CALL SEARCH(ICNOD,IPRESC,PRESKV,ISER,DEFMAX)
C-----
C      -- Setup of Prescribed Disp. for TREAC
C      WRITE(6,902) ISTAGE
C      902 FORMAT(' ', TRAC> Contact', I3)
C-----
C      CALL NULVEC(DREAC1,NB)

```

```

C-----
C      TRAC/1
C      -- TRAC Step 1: Prior Ratio Relation Calculation --
C      Hiroshi Nakashima
C      Transport Technology Research Laboratory
C      Carleton University
C-----
C      << Friction Calculation Included, No Inflation Process >>
C-----
C      Input Data File(1) --> 'gene3d.dat'
C      Input Data File(2) --> 'experm.dat'
C      Output Data File(1) --> 'geomnet.out'
C-----
C      (Ver. 1.0, June 10, 1991)
C-----
C      IMPLICIT REAL*8 (A-H,O-Z)
C      PARAMETER(NPMAX=700,NEMAX=400,NNODE=8,NNMAX=4)
C      PARAMETER(IKV=17000,ILOAD=NPMAX*3,IPRESC=100)
C      COMMON /COORD/ X(NPMAX),Y(NPMAX),Z(NPMAX)
C      COMMON /ENOD/ LNODE(NEMAX,NNODE)
C      COMMON /BOUND/ NBC(NPMAX,3)
C      COMMON /MATER/ NMAT(NEMAX),YOUNG(NNMAX),POISS(NNMAX)
C      COMMON /DISPL/ UX(NPMAX),UY(NPMAX),UZ(NPMAX)
C      COMMON /CONT/ ICONT(NPMAX)
C      COMMON /PARAM/ TA(3,10),N1,N2,N3
C      COMMON /REACT/ FX(NPMAX),FY(NPMAX),FZ(NPMAX)
C      COMMON /MEINF/ NP,NE,NM,NB
C      COMMON /GEOMS/ WIDTH(100),CONTL(100),WLOAD(100),DEFLS(100)
C      COMMON /BUFS/ GF(100),XG(100),VR(100)
C-----
C      CHARACTER*40 TITLE
C      REAL*8 BEE(6,24),BTDB(2,24),COORD(6,3),DEE(6,6)
C      REAL*8 DER(3,8),DERIV(3,8),DBEE(6,24),EKM(24,24)
C      REAL*8 ELD(24),SAMP(2,2)
C      REAL*8 TKV(IKV),TREAC(ILOAD),DREAC(ILOAD),DREAC1(ILOAD)
C      REAL*8 XJAC(3,3),XJAC1(3,3),SAMP1(2,2)
C      REAL*8 TKVIL(IPRESC,ILOAD),TKVIR(IPRESC)
C-----
C      INTEGER IG(NEMAX,24),KDIAG(ILOAD),ICNOD(IPRESC)
C      DATA ISAMP/2,IBTDB,IKM,IBT,IDOF/4*24,IDEE,IBEE,IBEE,IB/4*6/
C      DATA ICOORD,NOD/2*8,IJAC,IJAC1,IDER,IDERIV,NODOF,IT/6*3/
C      DATA IDR2/2/
C-----
C      Constants Set
C-----
C      EPSI=0.5D0
C      NGP=1
C      CALL GAUSS(SAMP1,ISAMP,NGP)
C      NGP=2
C      CALL GAUSS(SAMP,ISAMP,NGP)
C-----
C      Repetition Entry

```



```

      IF (RATIO.LT.1.0D-3) IEND=1
      IF (FRCFAC.EQ.0.0D+0) IEND=1
      IF (K1GT.50) CALL EMSGE(2,K1,10)
      C-----
      C      Update of Displacement Vector
      C-----
      CALL UPDISP(NB,TREAC,DREAC,K1)
      C-----
      C      Iterative Reaction at Contact Surface
      C-----
      DO 60 I=1,IPC
        XR=0.0D+0
        DO 65 K=1,NB
          XR=XR+TKVIL(I,K)*DREAC(K)
          RR=XR-TKVIR(I)
          J=ICNOD(I)
          J2=NBC(J,2)
          J1=NBC(J,1)
          J3=NBC(J,3)
          DREAC1(J2)=RR
          C-----
          C      --- FRCFAC: Coulomb Friction Factor
          IF (J1.NE.0) THEN
            IF (FRCFAC.NE.0.0D+0) THEN
              XX=TANH(DABS(TREAC(J1))/EPSI)
              DREAC1(J1)=FRCFAC*RR*XX*TREAC(J1)/DABS(TREAC(J1))
            ENDIF
          ENDIF
          IF (J3.NE.0) THEN
            IF (FRCFAC.NE.0.0D+0) THEN
              XX=TANH(DABS(TREAC(J3))/EPSI)
              DREAC1(J3)=FRCFAC*RR*XX*TREAC(J3)/DABS(TREAC(J3))
            ENDIF
          ENDIF
        END DO
      END DO
      C-----
      C      IF (IEND.EQ.0) GOTO 700
      C-----
      C      Check of No Penetration
      C-----
      CALL PENETR(ICNOD,IPRESC,IPENE,PRESCV,TREAC)
      IF (IPENE.EQ.1) GOTO 600
      C-----
      C      Transform to Ux, Uy and Uz
      C-----
      DO 70 IE=1,NE
        K=0
        DO 75 M=1,IDO
          IF (IG(IE,M).EQ.0) ELD(M)=0.0E+0
          IF (IG(IE,M).NE.0) ELD(M)=TREAC(IG(IE,M))
          IF (MOD(M,3).EQ.0) THEN
            K=K+1
            I1=LNODE(IE,K)
          END IF
        END DO
      END DO
      C-----
      C      DO 70 IE=1,NE
      C-----
      C      DO 75 M=1,IDO
      C-----
      C      IF (IG(IE,M).EQ.0) ELD(M)=0.0E+0
      C-----
      C      IF (IG(IE,M).NE.0) ELD(M)=TREAC(IG(IE,M))
      C-----
      C      IF (MOD(M,3).EQ.0) THEN
      C-----
      C      K=K+1
      C-----
      C      I1=LNODE(IE,K)
      C-----
      C      SUBROUTINE SAVEGM(ISTAGE,KR)
      C-----
      C      STOP
      C-----
      C      IF (KR.EQ.1) GOTO 720
      C-----
      C      590 CALL SAVEGM(ISTAGE,KR)
      C-----
      C      GOTO 500
      C-----
      C      IF (TDEFL.GT.DEFMAX) GOTO 590
      C-----
      C      IF (ISTAGE.EQ.80) GOTO 590
      C-----
      C      End of Calculation Check by Stage & Deflection
      C-----
      CALL CONDAT(ISTAGE)
      C-----
      CALL SEARCH(ICNOD,IPRESC,PRESCV,ISER,DEFMAX)
      C-----
      C      Search for Adjacent Contact Nodes
      C-----
      C      950 FORMAT(' Current Deflection(cm) =',F8.4)
      C-----
      C      CALL LDSAVE(ISTAGE,ICNOD,IPC,PRESCV,IFY)
      C-----
      C      WRITE(6,950) TDEFL
      C-----
      C      WLOAD(ISTAGE)=WL
      C-----
      C      DEFLS(ISTAGE)=TDEFL
      C-----
      C      TDEFL=TDEFL+PRESCV
      C-----
      C      80 CONTINUE
      C-----
      C      WL=WL+FY(J)
      C-----
      C      J3=NBC(J,3)
      C-----
      C      J1=NBC(J,1)
      C-----
      C      J2=NBC(J,2)
      C-----
      C      J=ICNOD(I)
      C-----
      C      DO 80 I=1,IPC
      C-----
      C      WL=0.0D0
      C-----
      C      WRITE(' ') 'REACT'
      C-----
      C      C      Load-Deflection Save
      C-----
      C      CALL UPDATE
      C-----
      C      CALL DSAVE(ISTAGE,ITITLE)
      C-----
      C      WRITE(6,' ') 'DISP Save'
      C-----
      C      Result Output & Coord Update
      C-----
      70 CONTINUE
      C-----
      75 CONTINUE
      C-----
      ENDIF
      UZ(I1)=ELD(M )
      UY(I1)=ELD(M+1)
      UX(I1)=ELD(M+2)

```

```

C      This subroutine saves the contact geometry, such as deflections,
C      contact lengths and contact widths, and normal reactions for
C      each Young's modulus ratio selected.
C
C      geomet1.out and geomet2.out are final outputs.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*80 IDFILE
      COMMON /GEOMS/ WIDTH(100),CONTL(100),WLOAD(100),DEFLS(100)
C
C      WRITE(IDFILE,920) KR
920 FORMAT('geomet',I1,'.out')
      OPEN(16,FILE=IDFILE)
      WRITE(16,'1') ISTAGE
      DO 10 I=1,ISTAGE
10    WRITE(16,900) DEFLS(I),WIDTH(I),CONTL(I),WLOAD(I)
900 FORMAT(4F15.7)
      WRITE (16,910)
910 FORMAT(1X,'Deflection (cm)',1X,'Contact width (cm)',
      & 1X,'Contact length (cm)',1X,'Normal reaction (N)')
      CLOSE(16)
      RETURN
      END

```

```

C-----
C      TRAC/M11
C      -- TRAC Module 11: Search Program for Modulus Ratio --
C      Hiroshi Nakashima
C      Transport Technology Research Laboratory
C      Carleton University
C
C      < Calculation based on Projected Area >
C      < Golden Section-ratio Search & Least Square Method >
C
C      Input Data File(1) -> 'geomat.out'
C      Input Data File(2) -> 'experim_x.dat'
C      Output Data File(1) -> 'ratio.dat'
C      Output Data File(2) -> 'ratio.org'
C
C      (Ver. 1.2, June 28, 1991)
C-----
C      AR(*) : Coefficients for Least Square Equation
C      NAR   : Order of Coefficients (= N+1)
C-----
C      IMPLICIT REAL*8 (A-H,O-Z)
C      COMMON /PARAM/ TA(10),NX
C      COMMON /PARAG/ AR(10),NAR
C      COMMON /LINES/ AX(80),BX(80),FLAG(80)
C      COMMON /DATAF/ X1(80,2),Y1(80,2),Y2(80,2),LM(2)
C      COMMON /COORD/ XC(80),NC
C      COMMON /PNTDD/ XP(50),PX(50),NP
C      COMMON /BUFFS/ GF(100),XG(100),VR(100)
C      COMMON /L SMP/ TB(10),NB
C
C      REAL*8 RATD(2)
C
C      RATD(*) : Ratio Value used in 'trac.f'
C
C      RATD(1)=1.0D0
C      RATD(2)=5.0D0
C
C      WRITE(6,*) '*** T R A C/M11-Search in Process ***'
C
C      get experimental contact area-deflection relationship
C
C      WRITE(6,*) 'PARAMS'
C      CALL PARAMS
C
C      get GEOMET1.DAT and GEOMET2.DAT and process them
C
C      WRITE(6,*) 'LDGEOM'
C      CALL LDGEOM
C      --- J-loop for two data files
C      DO 20 J=1,2
C      --- I-loop for Contact Width and Length
C      DO 10 I=1,2

```

```

C
C      select some of the points from GEOMET1.DAT
C      and GEOMET2.DAT
C
C      CALL VSETS(I,J)
C
C      get regression coefficients for contact width
C      and contact length
C
C      CALL REGS(0)
C
C      use the new polynomials to get two sets of
C      contact width and contact length data
C
C      DO 16 I1=1,LM(J)
C      --- XX is deflection
C      XX=X1(I1,J)
C      CALL RVALUE(XX,TB,NB,YRR)
C
C      IF (I.EQ.1) THEN
C      --- YRR is contact width
C      Y1(I1,J)=YRR
C      ELSE
C      --- YRR is contact length
C      Y2(I1,J)=YRR
C      ENDIF
C
C      16 CONTINUE
C      10 CONTINUE
C      20 CONTINUE
C
C      calculate linear coefficients for contact area-deflection line
C
C      WRITE(6,*) 'LINEAR'
C      CALL LINEAR
C
C      conduct
C
C      WRITE(6,*) 'SEARCH'
C      CALL SEARCH
C      WRITE(6,*) 'PICK'
C      CALL PICK(RATD)
C      WRITE(6,*) 'REGS'
C      CALL REGS(1)
C      STOP
C      END
C-----
C      S:JROUTINE LINEAR
C
C      This subroutine calculates contact area and
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      COMMON /DATAF/ X1(80,2),Y1(80,2),Y2(80,2),LM(2)

```

```

COMMON /LINES/ AX(80),BX(80),IFLAG(80)
C
DO 100 L=1,LM(1)
C
C calculate contact area
C
Y11=Y1(L,1)*Y2(L,1)
Y12=Y1(L,2)*Y2(L,2)
C
C calculate the linear coefficients for this line
C
X11=X1(L,1)
X12=X1(L,2)
DY=Y11-Y12
DX=X11-X12
IFLAG(L)=0
IF (DABS(DX).LT.1.0D-10) THEN
C
C take care of the case when the line is vertical
C
IFLAG(L)=1
AX(L)=1.0D0
BX(L)=0.0D0
GOTO 100
ENDIF
DXY=Y11*X12-Y12*X11
C
C save the coefficients in AX and BX
C
AX(L)=DY/DX
BX(L)=DXY/DX
100 CONTINUE
RETURN
END
C.....
SUBROUTINE PICK(RATD)
C
C This subroutine interpolates the values of alpha
C
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /DATA/ X1(80,2),Y1(80,2),X2(80,2),LM(2)
COMMON /LINES/ AX(80),BX(80),IFLAG(80)
COMMON /COORD/ XC(80),NC
COMMON /PNTDD/ XP(50),RX(50),NP
REAL*8 RATD(*)
C
J=0
DO 100 I=1,LM(1)
C
C -- XX is deflection at the measured
C area deflection curve
C
XX=XC(I)
C
C -- XR1 is deflection at the curve alpha = 1
C XR2 is deflection at the curve alpha = 5
C
XR1=X1(I,1)
XR2=X1(I,2)
IF ((IFLAG(I).EQ.1) .OR. (I.EQ.1)) THEN
C
C if the line is vertical or the first point,
C calculate the contact area, and use it to
C estimate alpha
C
YR1=Y1(I,1)*Y2(I,1)
YR2=Y1(I,2)*Y2(I,2)
DY=YR2-YR1
RX=DX*DABS(RATD(2)-RATD(1))+RATD(1)
IF ((RX.LE.0.D0) .OR. (XR1.LT.0.D0)) GOTO 100
RX(J)=RX
XP(J)=XR1
GOTO 100
ENDIF
C
C use the proposed linear relationship to
C estimate alpha
C
DX=(XR1-XX)/(XR1-XR2)
RX=DX*DABS(RATD(2)-RATD(1))+RATD(1)
C
C take care of rare cases
C
IF ((RX.LE.0.D0) .OR. (XX.LT.0.D0)) GOTO 100
C
C RX alpha values
C XP deflections
C
RX(J)=RX
XP(J)=XX
100 CONTINUE
NP=J
C
C -- RE-ARRANGING OF X
C sort XP in ascending order
C
DO 300 I=1,NP-1
DO 300 J=I+1,NP
IF (XP(I).GT.XP(J)) THEN
XPB=XP(I)
RXB=RX(I)
XP(I)=XP(J)
RX(I)=RX(J)
XP(J)=XPB
RX(J)=RXB
ENDIF
300 CONTINUE
C
C DO 505 I=1,NP

```



```

C
C      This subroutine calculates the error between the true contact area
C      from the area-deflection curve and the predicted one
C
C      IMPLICIT REAL*8 (A-H,O-Z)
COMMON /LINES/ AX(80),BX(80),IFLAG(80)
C
C      WW : deflection
C      WE : predicted contact area
C      WGDC : true contact area
C
C      WB=AX(I)*WW+BX(I)
C      write(6,*) 'WB,WGDC,WW=',WB,WGDC,WW
C      FUNCS=(WB-WGDC)*(WB-WGDC)
C      RETURN
C      END
C
C      SUBROUTINE RVALUE(WW,TA,NX,WR)
C
C      This subroutine calculates the value of a polynomial using
C      the newly obtained regression coefficients
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      REAL*8 TA(*)
C
C      WW is deflection
C      TA is coefficient
C      WR is function value
C      NX is the order
C
C      WR=TA(NX)
C      DO 10 IT=2,NX
C        J=NX-IT+1
C        WR=WR+TA(J)*WW**((IT-1))
C      10 CONTINUE
C      RETURN
C      END
C
C      SUBROUTINE PARAMS
C
C      This subroutine reads the contact area-deflection relationship
C      from static test results
C
C      IMPLICIT REAL*8 (A-H,O-Z)
COMMON /PARAM TA(10),NX
C      REAL*8 TX(10)
C
C      OPEN(18,FILE='experiment 1.dat')
C      READ(18,*) NX
C      READ(18,*) (TA(K),K=1,NX)
C      READ(18,*) NDUMMY
C      READ(18,*) (TX(K),K=1,NDUMMY)
C
C      CLOSE(18)
C      RETURN
C      END
C
C      SUBROUTINE GSFUNC(XV1,XV2,GFUNC,KP,LRMOD)
C
C      This subroutine performs a golden-section search
C
C      IMPLICIT REAL*8 (A-H,O-Z)
COMMON /BUFFS/ GF(100),XG(100),VR(100)
C
C      RR=0.6180339887D0
C      KIND=MOD(KP,2)
C      IF (KIND.EQ.0) THEN
C        GFUNC=XV1+RR*(XV2-XV1)
C        GF(KP)=GFUNC
C      ELSEIF (KIND.EQ.1) THEN
C        GFUNC=XV2-RR*(XV2-XV1)
C        GF(KP)=GFUNC
C      ENDIF
C      IF ((LRMOD.EQ.1) .AND. (KIND.EQ.0)) GFUNC=GF(KP-3)
C      IF ((LRMOD.EQ.2) .AND. (KIND.EQ.1)) GFUNC=GF(KP-1)
C      WRITE(6,*) 'KP,GFUNC=',KP,GFUNC
C      RETURN
C      END
C
C      SUBROUTINE LDGEOM
C
C      This subroutine reads and processes the output data files from
C      TRAC1, ie. GEOMET1 OUT and GEOMET2 OUT
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      CHARACTER*80 IDFILE
COMMON /DATAF, X1(80,2),Y1(80,2),Y2(80,2),LM(2)
C      REAL*8 WIDTH(100),CONTL(100),DEFLS(100)
C
C      DO 5 K=1,2
C
C        WRITE(IDFILE,991) K
C991  FORMAT('geom',1,'.out')
C
C        OPEN(16,FILE=IDFILE)
C        READ(16,*) ISTAGE
C
C        DEFLS deflection
C        WIDTH contact width
C        CONTL contact length
C        WX normal reaction
C
C        DO 10 I=1,ISTAGE
C          READ(16,*) DEFLS(I),WIDTH(I),CONTL(I),WX
C          CLOSE(16)

```



```

      READ(5,*) NAR
      ENDIF
      DO 100 I=1,NP
        W(I)=1.0D0
      100 CONTINUE
      C
      C get coefficient matrix A, which is NP by NAR
      C or NPT by NPM
      C
      CALL COEFF(XP,A,NP,NAR)
      C
      C get the final coefficient matrix AWA, which
      C is NAR by NAR, and the final vector AR, which
      C is NAR by 1
      C
      CALL GENER(A,W,RX,NP,NAR,AWA,AR,50,25)
      C
      C solve the simultaneous linear algebraic equations
      C
      CALL GAUSSE(AWA,AR,25,NAR)
      C
      IF (IMODE.EQ.1) THEN
        ... IMODE = 1, alpha fit
        CALL SAVEPM(AR,NAR)
        CALL DATASS(XP,RX,NP)
      ELSE
        ... IMODE = 0, contact width or
        contact length fit
        CALL PARAX(AR,NAR)
      ENDIF
      RETURN
      END
      C
      SUBROUTINE DATASS(XP,RX,NP)
      C
      C This subroutine saves the data points used for obtaining the
      C polynomial coefficients in: file RATIO.DAT
      C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 XP(*),RX(*)
      C
      OPEN(12,FILE='ratio.org')
      WRITE(12,*) NP
      DO 10 I=1,NP
        WRITE(12,900) XP(I),RX(I)
      10 CONTINUE
      900 FORMAT(2F15.7)
      CLOSE(12)
      RETURN
      END
      C
      SUBROUTINE SAVEPM(AR,NAR)
      C
      C This subroutine saves the final regression coefficients
      C in file RATIO.DAT
      C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 AR(*)
      C
      WRITE(6,*) '..... RESULT OF REGRESSION .....'
      OPEN(10,FILE='ratio.dat')
      WRITE(10,*) NAR
      WRITE(10,*) (AR(I),I=NAR,1,-1)
      DO 10 I=NAR,1,-1
        WRITE(6,990) I,AR(I)
      10 CONTINUE
      990 FORMAT(5X,I3,2X,E15.7)
      CLOSE(10)
      RETURN
      END
      C
      SUBROUTINE PARAX(AR,NAR)
      C
      C This subroutine assigns the regression coefficients in AR
      C to TB and reveals the order
      C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /LSMPP/ TB(10),NB
      REAL*8 AR(*)
      C
      NB=NAR
      L=0
      DO 5 K=NB,1,-1
        L=L+1
        TB(L)=AR(K)
      5 CONTINUE
      RETURN
      END
      C
      SUBROUTINE COEFF(X,A,NPT,NPM)
      C
      C This subroutine calculates the coefficients matrix A(I,J)
      C A is a NPT by NPM matrix, where NPT is number of data points
      C and NPM is number of coefficients
      C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 X(*),A(50,*)
      C
      C X deflection
      C
      DO 10 I=1,NPT
        DO 20 J=1,NPM
          IF (J.EQ.1) THEN

```



```

      A(I,J)=1.0D+0
    ELSE
      AX=1.0D+0
      DO 30 K=2,J
        AX=AX*X(I)
      30  A(I,J)=AX
      ENDIF
    20  CONTINUE
    10  CONTINUE
    RETURN
  END
C-----
C  SJBROUTINE GENERI(A,W,B,NPT,NPM,AA,BB,NA,NB)
C
C  This subroutine obtains the matrix, AA, at left-hand side and
C  the vector, BB, at right-hand side before solving
C  the linear equations
C
C  IMPLICIT REAL*8 (A-H,O-Z)
C  REAL*8 A(NA,*),W(*),B(*),AA(NB,*),BB(*),AT(25,25)
C
C  get the transpose of matrix A, and store in matrix AT.
C  AT is a NPM by NPT matrix, where NPM is number of
C  coefficients, and NPT is number of data points
C
C  DO 10 I=1,NPM
C  DO 10 K=1,NPT
    AT(I,K)=A(K,I)*W(K)
  10  CONTINUE
C
C  W : unit vector
C  B : contact length, or contact width or contact area
C  AA : final matrix at left-hand side
C  BB : final vector at right-hand side
C
C  AA = AT A
C  BB = AT B
C
C  DO 20 I=1,NPM
    SUM=0.0D+0
    DO 30 J=1,NPM
      SUM1=0.0D+0
      DO 35 K=1,NPT
        SUM1=SUM1+AT(I,K)*A(K,J)
      35  CONTINUE
      AA(I,J)=SUM1
    30  CONTINUE
C
C  DO 40 K=1,NPT
    SUM=SUM+AT(I,K)*B(K)
  40  CONTINUE
    BB(I)=SUM
  END

```

```

20 CONTINUE
RETURN
END
C-----
C  SUBROUTINE GAUSSE(A,B,NM,N)
C
C  This subroutine solves a set of simultaneous linear algebraic
C  equations using Gaussian elimination method
C
C  IMPLICIT REAL*8 (A-H,O-Z)
C  REAL*8 A(NM,*),B(*)
C
C  NM1=N-1
C  DO 30 I=1,NM1
    XLARGE=1 D-15
    JCOUNT=0
    DO 15 J=I,N
      IF (DABS(A(J,I)) GT XLARGE) THEN
        XLARGE=DABS(A(J,I))
        JCOUNT=J
      ENDIF
    15  CONTINUE
    IF (JCOUNT.EQ.0) GOTO 60
    DO 20 K=I,N
      TEMP=A(I,K)
      A(I,K)=A(JCOUNT,K)
      A(JCOUNT,K)=TEMP
    20  CONTINUE
    TEMP=B(I)
    B(I)=B(JCOUNT)
    B(JCOUNT)=TEMP
    IP1=I+1
    DO 30 K=IP1,N
      Q=A(K,I)/A(I,I)
      B(K)=Q*B(I)+B(K)
      DO 30 J=IP1,N
        A(K,J)=Q*A(I,J)+A(K,J)
      30  CONTINUE
    IF (A(N,N).EQ.0.0D+0) GOTO 60
    B(N)=B(N)/A(N,N)
C
    NP1=N+1
    DO 50 K=1,NM1
      Q=0.0D+0
      NMK=N-K
      DO 40 J=1,K
        Q=Q+A(NMK,NP1-J)*B(NP1-J)
      40  CONTINUE
      B(NMK)=(B(NMK)-Q)/A(NMK,NMK)
    50  CONTINUE
    RETURN
C

```

```
60 WRITE(6,900) ..... Error : Matrix is Singular. ....//  
900 FORMAT(/, .....//  
STOP  
RETURN  
END
```

```

C.....
C      TRAC/2
C      .. TRAC Step 2 : Young's Modulus Search ..
C      Hiroshi Nakashima
C      Transport Technology Research Laboratory
C      Carleton University
C
C      << Friction Calculation Included, No Inflation Process >>
C      << Young's Moduli are searched with 'ratio.dat' >>
C
C      Input Data File(1) --> 'gene3d.dat'
C      Input Data File(2) --> 'experim.dat'
C      Input Data File(2') --> 'experim_x.dat'
C      Input Data File(3) --> 'ratio.dat'
C      Output Data File(1) --> 'youngs.prm'
C      Output Data File(2) --> 'numgdc.dat'
C
C      (Ver 1.5, July 10, 1991)
C.....
C      IMPLICIT REAL*8 (A-H,O-Z)
C      PARAMETER(NPMAX=700,NEMAX=400,NNODE=8,NIMMAX=4)
C      PARAMETER(IKV=340000,ILOAD=NPMAX*3,IPRESC=100)
C      COMMON /COORD/ X(NPMAX),Y(NPMAX),Z(NPMAX)
C      COMMON /ELNOD/ LNODE(NEMAX,NNODE)
C      COMMON /BOUND/ NBC(NPMAX,3)
C      COMMON /MAT/ NMAT(NEMAX),YOUNG(NMMAX),POISS(NIMMAX)
C      COMMON /DISPL/ UX(NPMAX),UY(NPMAX),UZ(NPMAX)
C      COMMON /CONTC/ ICNT(NPMAX)
C      COMMON /PARAM/ TA(3,10),N1,N2,N3
C      COMMON /REACT/ FX(NPMAX),FY(NPMAX),FZ(NPMAX)
C      COMMON /MEINF/ NP,NE,NIM,NB
C      COMMON /GEOMS/ WIDTH(120),CONTL(120),WLOAD(120)
C      COMMON /PARAR/ TAR(10),NR
C      COMMON /BUFFS/ GF(100),XG(100),VR(100)
C
C      CHARACTER*40 ITITLE
C      REAL*8 BEE(6,24),BTDB(24,24),COORD(8,3),DEE(6,6)
C      REAL*8 DER(3,8),DERIV(3,8),DBEE(6,24),EKM(24,24)
C      REAL*8 ELD(24),SAMP(2,2)
C      REAL*8 TKV(IKV),TREAC(ILOAD),DREAC(ILOAD),DREAC(1,LOAD)
C      REAL*8 XJAC(3,3),XJAC1(3,3),SAMP1(2,2),YRATIO(100)
C      REAL*8 TKVIL(IPRESC,LOAD),TKVIR(IPRESC),BDVAL(2)
C      REAL*8 YGM(100),DEFLS(100)
C
C      INTEGER IG(NEMAX,24),KDIAG(ILOAD),ICNOD(IPRESC)
C      DATA ISAMP/2/ ,BTDB,IKM,IBT,IDOFL/4/24,IDEE,IBEE,IDBEE,IH/4/6/
C      DATA ICOORD,HOD/2/ ,IAC,IAC1,IDER,IDERIV,NODOF,I/6/3/
C      DATA IDER2/
C.....
C      Parameter Setup
C.....
C      ISTAGE=0

```

```

NGP=1
CALL GAUSS(SAMP1,ISAMP,NGP)
NGP=2
CALL GAUSS(SAMP,ISAMP,NGP)
TDEFL=0.0D+0
TFY=0.0D+0
EPSI=0.5D0
CALL NULVEC(FX,NPMAX)
CALL NULVEC(FY,NPMAX)
CALL NULVEC(FZ,NPMAX)
C      --- Parameter Search Boundary Data
C      BDVAL(1)=1.0D-5/10.
C      BDVAL(2)=2.0D-6/10
C.....
C      Data Input
C.....
C      WRITE(6,*) '*** T R A C / 2 in Process ***'
C      WRITE(6,*) 'DATAIN'
C      CALL DATAIN(TITLE,PRESS,FRCFAC,IP1,DEFMAX)
C      CALL PARAMS
C      CALL RATIOS
C      WRITE(6,*) 'DEFMAX= ',DEFMAX
C      CALL SETBC
C.....
C      Calculation of KDIAG
C.....
C      DO 10 I=1,NB
C        KDIAG(I)=0
C      10 CONTINUE
C      DO 15 IE=1,NE
C        CALL IGSET(IE,IG)
C        CALL FKDIAG(KDIAG,IG,IDOFL,IE,NEMAX)
C      15 CONTINUE
C      KDIAG(1)=1
C      DO 20 I=2,NB
C        KDIAG(I)=KDIAG(I)+KDIAG(I-1)
C      20 CONTINUE
C      NN=KDIAG(NB)
C      IF (NN.GT.IKV) CALL EMSSGE(1,NN,IKV)
C.....
C      Loop Entry of Incremental Contact
C.....
C      500 JCONTINUE
C      ISTAGE=ISTAGE+1
C.....
C      Loop Entry of Penetration & Param Search
C.....
C      KP=0
C      IMODE=0
C      600 CONTINUE
C      CALL NULVEC(TREAC,NB)
C.....

```

```

IF (ISTAGE.EQ.1) CALL SEARCH(ICNOD,IPRESC,PRESKV,ISER,DEFMAX)
C-----
C *** Parameter Search Setup ***
C-----
IF (IMODE.EQ.3) GOTO 977
IMODE=3
C
979 KP=KP+1
TDX=TDEFI+PRESCV
WRITE(6,975) KP
975 FORMAT(' ',++ E_1 Parameter Search Step: ',I3,' +++++')
IF (KP.LE.2) THEN
LRMOD=0
XV1=BDVAL(1)
XV2=BDVAL(2)
ENDIF
CALL GSFUNC(XV1,XV2,XXV,KP,LRMOD)
XG(KP)=XXV
C
CALL VLRT(ERATIO,TDEFI)
YOUNG(2)=ERATIO*XXV
YOUNG(1)=XXV
C
write(' ', 'KPE_1.E_2=',kp,young(1),young(2))
IF ((LRMOD.EQ.1) AND (MOD(KP,2).EQ.0)) THEN
XG(KP)=XXV
GOTO 557
ELSEIF ((LRMOD.EQ.2) AND (MOD(KP,2).EQ.1)) THEN
XG(KP)=XXV
GOTO 558
ENDIF
977 IF (IMODE.EQ.3) THEN
WRITE(6,' ') *** Final Calculation in Current Stage ***
YRATIO(ISTAGE)=ERATIO
YOUNG(2)=ERATIO*YGM(ISTAGE)
YOUNG(1)=YGM(ISTAGE)
ENDIF
C-----
C *** Setup of Prescribed Disp for TREAC
C-----
WRITE(6,902) ISTAGE
902 FORMAT(' ',TRAC> Contact: ',I3)
C
CALL NULVEC(DREAC,NB)
C
IPC=0
DO 25 I=1,NP
IF (ICONT(I).EQ.1) THEN
IPC=IPC+1
ICNOD(IPC)=I
J=NBC(I,2)
TREAC(J)=PRESCV
ENDIF
25 CONTINUE
C
WRITE(6,895) PRESCV
895 FORMAT(' [Prescribed Nodes] with Displ.= ',F8.4)
WRITE(6,962) (ICNOD(I),I=1,IPC)
962 FORMAT(3X,8I6)
IPENE=0
KI=0
C-----
C Newton-Raphson Friction Loop
C-----
700 CONTINUE
CALL NULVEC(TKV,NN)
KI=KI+1
WRITE(6,903) KI
903 FORMAT(' --[NR Loop: ',I2,']-----')
CALL NULVEC(DREAC,NB)
C
IF (KI.GT.1) THEN
DO 30 I=1,IPC
J2=NBC(ICNOD(I),2)
DREAC(J2)=0.0D+0
30 CONTINUE
ENDIF
XX=0.0D+0
DO 35 I=1,NB
XX=XX+TREAC(I)*TREAC(I)
35 CONTINUE
VNORM=DSORT(XX)
C-----
C Element Stiffness Integration and Assembly
C
ISTAGE=1 Non-iterative solution
C
ISTAGE>1: Newton-Raphson iteration
C-----
WRITE(6,' ') STIFF
DO 40 IE=1,NE
CALL ECOORD(IE,COORD,ICOORD)
CALL NULL(EKM,IKM,IDO,IDO)
CALL MATPRONMAT(IE),TDEFI,EX)
E=YOUNG(INMAT(IE))
V=POISS(INMAT(IE))
CALL FORMD3(DEE,DEE.E,V)
-- Gauss Integration Loop
DO 45 I=1,NGP
DO 45 J=1,NGP
DO 45 K=1,NGP
-- Stiffness Matrix
CALL FMING(IDER,IDER,SAMP,ISAMP,I,J,K)
CALL MATMUL(IDER,IDER,COORD,ICOORD,XJAC,IJAC,IT,NOD,IT)
CALL TREEX3(XJAC,IJAC,XJAC1,IJAC1,DET,IE)
CALL MATMUL(XJAC1,IJAC1,DER,IDER,DERIV,IDER,IT,IT,NOD)

```

```

CALL NULL(BEE,IBEE,IH,IDOF)
CALL FORMB3(BEE,IBEE,DERIV,IDERIV,NOD)
CALL MATMUR(DEE,IDEE,BEE,DBEE,IBDDEE,IH,IH>IDOF)
QUOT=DET*SAMP(I,2)*SAMP(J,2)*SAMP(K,2)
CALL BTDBCC(BTD8,BEE,DBEE,IBTD8,IBDDEE,QUOT)
CALL MATADDC(EKM,KIM,BTD8,IBTD8>IDOF>IDOF)
45 CONTINUE
CALL FSPARV(TKV,EKM,IKM,IG,KDIAG>IDOF,I,E,NEMAX)
40 CONTINUE
C-----
C Right-hand side Arrangement
C-----
CALL LINML5(TKV,TREAC,DREAC,NB,KDIAG)
CALL VEC SUB(DREAC,DREAC1,DREAC,NB)
C-----
C Prescribed Displacement Condition
C-----
DO 50 I=1,IPC
I1=ICNOD(I)
J=NBC(I,2)
CALL PRESCDI,J,KDIAG,NB,TKV,DREAC,TKVIL,TKVIR,ILOAD,*
0.0D+0,IPRESO)
50 CONTINUE
C-----
C Solve Linear Equations by Skyline Method
C-----
WRITE(6,*) 'SPARIN'
CALL SPARIN(TKV,NB,KDIAG)
WRITE(6,*) 'SPABAC'
CALL SPABAC(TKV,DREAC,NB,KDIAG)
C-----
C Check of Convergence
C-----
IEND=0
XX=0.0D+0
DO 55 I=1,NB
XX=XX+DREAC(I)*DREAC(I)
55 CONTINUE
VNORM1=DSQRT(XX)
RATIO=VNORM1/VNORM
WRITE(6,907) K1,RATIO
907 FORMAT(' -[Norm](',I2,'):',F10.5,'] -----')
IF (RATIO.LT.1.0D-3) IEND=1
IF (FRCFAC.EQ.0.0D+0) IEND=1
IF (K1.GT.50) CALL EMSSGE(2,K1,50)
C-----
C Update of Displacement Vector
C-----
CALL UPDISP(NB,TREAC,DREAC,K1)
C-----
C Iterative Reaction at Contact Surface
C-----

```

```

CX  WRITE(' ') 'REACT-PRE'
    FLOAD=0.0D+0
    DO 81 I=1,IPC
        J=ICNOD(I)
        J2=NBC(J,2)
        IF (J2.NE.0) FLOAD =DREAC1(J2)+FLOAD
    81 CONTINUE
C-----
C ***** Parameter Search for E_1 and E_2 *****
C
C  Function=(Witem*Wgde)*2, W=Load (kN)
C-----
    IF (IMODE.GT.0) THEN
        WX=(TFY+FLOAD*4.0)/1000.
        write(6,' ') 'Total Load=',WX
        CALL EVALU(IMODE,WX,TDX,FUNCS,WR)
        VR(KP)=FUNCS
        GOTO 556
    557 VR(KP)=VR(KP-3)
        GOTO 556
    558 VR(KP)=VR(KP-1)
    556 CONTINUE
        write(6,' ') kp,vr(kp)
        IF (DABS(FUNCS) LT .1D-5) GOTO 771
        IF (MOD(KP,2) EQ 1) GOTO 979
        VCHECK=VR(KP-1)-VR(KP)
        IF (VCHECK LT -.2D-3) THEN
            XV1=XV1
            XV2=XG(KP)
            LRMOD=1
            GOTO 979
        ELSEIF (VCHECK GT 2.0D-3) THEN
            XV1=XG(KP-1)
            XV2=XV2
            LRMOD=2
            GOTO 979
        ENDIF
    771 IF ((KP EQ 50) OR (DABS(FUNCS) LT .1D-5)
        .OR (DABS(VCHECK) LE 2.0D-3)) THEN
        IF (DABS(FUNCS) LT .1D-5) THEN
            YGM(ISTAGE)=XG(KP)
        ELSE
            YGM(ISTAGE)=(XG(KP)+XG(KP-1))*0.5D0
        ENDIF
    949 WRITE(6,949) YGM(ISTAGE)*10.D0
        FORMAT(' +++ Result of Search. E_1-->:F12.5,(kPa)')
        IMODE=3
        GOTO 977
    ENDIF
    END*
C-----
C  Result Output & Coord Update
C-----
C  WRITE(6,' ') 'DISP Save'
C  CALL DSAVE(ISTAGE,TITLE)
C  CALL UPDATE
C-----
C  Load-Deflection Save
C-----
    WRITE(' ') 'REACT'
    WX=0.0D0
    DO 80 I=1,IPC
        J=ICNOD(I)
        J2=NBC(J,2)
        J1=NBC(J,1)
        J3=NBC(J,3)
        IF (J1.NE.0) FX(J)=DREAC1(J1)+FX(J)
        IF (J2.NE.0) FY(J)=DREAC1(J2)+FY(J)
        IF (J3.NE.0) FZ(J)=DREAC1(J3)+FZ(J)
        WX=WX+DREAC1(J2)
    80 CONTINUE
C
    TDEFL=TDEFL+PRESCV
        <Full model>=4.0*<Quater model>
    TFY=TFY+WX*4.D0
    WLOAD(ISTAGE)=TFY
    DEFLS(ISTAGE)=TDEFL
    WRITE(6,950) TDEFL,TFY/1000.D0
    950 FORMAT(' Deflection(cm) =',F8.4,' Load(kN) =',F8.4)
C-----
C  Search for Adjacent Contact Nodes
C-----
    CALL SEARCH(ICNOD,IPRESC,PRESCV,ISER,DEFMAX)
    CALL CONDAT(ISTAGE)
C-----
C  End of Calculation Check by Stage & Deflection
C-----
    IF (ISTAGE EQ 120) GOTO 590
    IF (TDEFL GT DEFMAX) GOTO 590
    590 CALL SAVER(WLOAD,YRATIO,YGM,ISTAGE)
    CALL SAVEGM(DEFLS,WIDTH,CONTL,WLOAD,ISTAGE)
C
    STOP
    END
C-----
C  SUBROUTINE SAVER(WLOAD,YRATIO,YGM,ISTAGE)
C
C  This subroutine saves the loads and Youngs' moduli in file
C  'Youngs.prm'
C
C  IMPLICIT REAL*8 (A-H,O-Z)
C  REAL*8 WLOAD(' '),YGM(' '),YRATIO(' ')
C

```

```

OPEN(16,FILE=youngs.prm)
WRITE(16,'') ISTAGE
DO 20 I=1,ISTAGE
  YOUNG2=YGM(I)*YRATIO(I)**10.0
  WLOADX=WLOAD(I)/1000
  WRITE(16,900) WLOADX,YGM(I)**10.0,YOUNG2
20 CONTINUE
900 FORMAT(3F15.5)
CLOSE(16)
RETURN
END

C-----
SUBROUTINE SAVEGMD(DEFLS,WIDTH,CONTL,WLOAD,ISTAGE)
C
C This subroutine saves the contact geometry, such as deflections,
C contact lengths and contact widths, and normal reactions in file
C "trac2 geom"
C
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 DEFLS(*),WIDTH(*),CONTL(*),WLOAD(*)
C
OPEN(16,FILE='trac2 geom')
WRITE(16,'') ISTAGE
DO 10 I=1,ISTAGE
  WRITE(16,900) DEFLS(I),WIDTH(I),CONTL(I),WLOAD(I)
10 CONTINUE
900 FORMAT(4F15.5)
CLOSE(16)
RETURN
END

C-----
SUBROUTINE EVALU(IMODE,WW,TDEFL,FUNCS,WR)
C
C This subroutine calculates the relative error between the calculated
C normal reaction and applied normal load.
C
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /PARAM/ TA(3,10),N1,N2,N3
C
NX=N1
IF (IMODE.EQ.2) NX=N2
IF (IMODE.EQ.3) NX=N3
WGDC=TA(IMODE,NX)
IF ((IMODE.EQ.3) AND. (TDEFL.LE.1.0) AND. (N3.EQ.2)) THEN
  DDX=TA(IMODE,1)+TA(IMODE,2)
  WGDC=DDX*TDEFL
ELSE
  DO 10 I=2,NX
    J=NX-I+1
    WGDC=WGDC+TA(IMODE,J)*TDEFL**(I-1)
10 CONTINUE
ENDIF

```

```

WRITE(6,'') WW,WGDC,TDEFL=' ',WW,WGDC,TDEFL
WR=WGDC
FUNCS=(WW*WGDC)/(WW*WGDC)
RETURN
END

C-----
SUBROUTINE VLRTAT(YRATIO,TDX)
C
C This subroutine calculates the alpha-ratio according to the current
C deflection, and the polynomial obtained from TRACM11
C
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /PARAM/ TAR(10),NR
C
YRATIO=TAR(NR)
DO 10 I=2,NR
  J=NR-I+1
  YRATIO=YRATIO+TAR(J)*TDX**(I-1)
10 CONTINUE
RETURN
END

C-----
SUBROUTINE PARAMS
C
C This subroutine reads the experimental relationship between contact
C area and deflection
C
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /PARAM/ TA(3,10),N1,N2,N3
C
500 WRITE(6,'') ' GDC Data File:'
WRITE(6,'') ' 1)ContactWidth,Length,Load 2)ContactArea,Load'
WRITE(6,'') ' --> Select Proper Number shown above.'
READ(5,'') INUM
IF (INUM.EQ.1) THEN
  OPEN(18,FILE='experim.dat')
  READ(18,'') N1
  READ(18,'') (TA(1,K),K=1,N1)
  READ(18,'') N2
  READ(18,'') (TA(2,K),K=1,N2)
  READ(18,'') N3
  READ(18,'') (TA(3,K),K=1,N3)
  CLOSE(18)
ELSEIF (INUM.EQ.2) THEN
  OPEN(18,FILE='experim_x.dat')
  READ(18,'') N1
  READ(18,'') (TA(1,K),K=1,N1)
  READ(18,'') N3
  READ(18,'') (TA(3,K),K=1,N3)
  CLOSE(18)
ELSE
  GOTO 500

```

```

C      ENDIF
C      RETURN
C      END
C-----
C      SUBROUTINE RATIOS
C      This subroutine reads the alpha-deflection relationship obtained
C      from TRACM11.
C      IMPLICIT REAL*8 (A-H,O-Z)
C      COMMON /PARAB/ TAR(10),NR
C
C      OPEN(18,FILE='rato.dat')
C      READ(18,*) NR
C      READ(18,*) (TAR(K),K=1,NR)
C      CLOSE(18)
C      RETURN
C      END
C-----
C      SUBROUTINE GSFUNC(XV1,XV2,GFUNC,KP,LRMOD)
C
C      Golden section search routine.
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      COMMON /BUFFS/ GF(100),XG(100),VR(100)
C
C      RR=0.6180339887D0
C      KIND=MOD(KP,2)
C      IF (KIND.EQ.0) THEN
C         GFUNC=XV1+RR*(XV2-XV1)
C         GF(KP)=GFUNC
C      ELSEIF (KIND.EQ.1) THEN
C         GFUNC=XV2-RR*(XV2-XV1)
C         GF(KP)=GFUNC
C      ENDIF
C      IF ((LRMOD.EQ.1) .AND. (KIND.EQ.0)) GFUNC=GF(KP,3)
C      IF ((LRMOD.EQ.2) .AND. (KIND.EQ.1)) GFUNC=GF(KP,1)
C      RETURN
C      END
C-----
C      SUBROUTINE CONDAX(ISTAGE,W1,W2)
C
C      This subroutine is never used
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      PARAMETER(NPMAX=700,NEMAX=400)
C      COMMON /COORDS/ X(NPMAX),Y(NPMAX),Z(NPMAX)
C      COMMON /CONT/ ICONT(NPMAX)
C      COMMON /AENF/ NP,NE,NM,NB
C      COMMON /DISPL/ UX(NPMAX),UY(NPMAX),UZ(NPMAX)
C
C      XCMAX=X(1)+UX(1)

```

```

ZCMAX=Z(1)+UZ(1)
DO 10 J=2,NP
  IF ((ICONT(J).EQ.-1) THEN
    IF ((X(J)+UX(J)).GT.XCMAX) XCMAX=X(J)+UX(J)
    IF ((Z(J)+UZ(J)).GT.ZCMAX) ZCMAX=Z(J)+UZ(J)
  ENDIF
10 CONTINUE
W1=2.D0*ZCMAX
W2=2.D0*XCMAX
RETURN
END

```



```

C .....
C      TRAC/M2
C -- TRAC Module 2: Least Square Method Program
C    for Modulus Calculation --
C      Hiroshi Nakashima
C      Transport Technology Research Laboratory
C      Carleton University
C
C      Input File Name--> 'Youngs.prm'
C      Output File Name--> 'Youngs.coef'
C
C      (Ver. 1.0   June 24, 1991)
C .....
C      NPT : Number of Data Points
C      NPM : Number of Coefficients = N+1
C      N   : Order of Regression
C .....
C      IMPLICIT REAL*8 (A-H,O-Z)
C      REAL*8 A(80,25),B(80),W(80),AWA(25,25),AWB(25),X(80),C(80)
C      REAL*8 PARAM(25,2),XB(80)
C      INTEGER NPS(2)
C
C      JL=0
C      WRITE(6,*) 'DATAIN'
C      CALL DATAIN(X,B,C,W,NPT,NPX,IFLAG,NPS)
C
C      500 CONTINUE
C      JL=JL+1
C      IF (JL.EQ.2) THEN
C        DO 10 I=1,NPT
C          IF (NPS(JL-1).EQ.6) X(I)=XB(I)
C            B(I)=C(I)
C        10 CONTINUE
C      ENDIF
C      NPX=NPS(JL)
C      WRITE(6,*) 'COEFF'
C      CALL COEFF(X,A,NPT,NPX)
C      WRITE(6,*) 'GENEIR'
C      CALL GENERA(W,B,NPT,NPX,AWA,AWB,80,25)
C      WRITE(6,*) 'GAUSSE'
C      CALL GAUSSE(AWA,AWB,25,NPX)
C
C      NPM=NPX
C      IF (NPX.EQ.11) NPM=?
C      DO 20 I=1,NPM
C        PARAM(I,JL)=AWB(I)
C      20 CONTINUE
C
C      IF (JL.EQ.1) GOTO 500
C      WRITE(6,*) 'DSAVE'

```

```

C      CALL DSAVE(PARAM,NPS)
C
C      STOP
C      END
C .....
C      SUBROUTINE COEFF(X,A,NPT,NPM)
C
C      This subroutine computes the values of given variables of
C      different order associated with the polynomial coefficients
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      REAL*8 X(*),A(80,*)
C
C      DO 10 I=1,NPT
C        IF (NPM.LE.10) THEN
C          DO 20 J=1,NPM
C            IF (J.EQ.1) THEN
C              A(I,J)=1.0D+0
C            ELSE
C              AX=1.0D+0
C              DO 30 K=2,J
C                AX=AX*X(I)
C                A(I,K)=AX
C              30 CONTINUE
C            A(I,J)=AX
C          ENDIF
C        20 CONTINUE
C      ELSEIF (NPM.EQ.11) THEN
C        DO 22 J=1,NPM-9
C          IF (J.EQ.1) THEN
C            A(I,J)=1.0D+0
C          ELSE
C            AX=1.0D+0
C            DO 31 K=2,NPM-9
C              AX=AX*X(I)
C              A(I,K)=AX
C            31 CONTINUE
C          A(I,J)=AX
C        22 CONTINUE
C      ENDIF
C      10 CONTINUE
C      RETURN
C      END
C .....
C      SUBROUTINE DSAVE(PARAM,NPS)
C
C      This subroutine saves the polynomial coefficients
C      in file 'YOUNGS.COEF'
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      REAL*8 PARAM(25,2)
C      INTEGER NPS(*)
C

```

```

C
OPEN('12.FILE=youngs.coef')
DO 10 J=1,2
WRITE(12,'') NPS(J)
NPM=NPS(J)
C
WRITE(6,940)
940 FORMAT(7X,'VALUES OF PARAMETERS')
IF (NPM.LE.10) THEN
WRITE(6,950) J,(PARAM(I,J),I=NPM,1,-1)
WRITE(12,951) (PARAM(I,J),I=NPM,1,-1)
ELSEIF (NPM.EQ.11) THEN
NPM=NPM-9
DO 15 I=NPX,1,-1
IF (I.EQ.1) PARAM(I,J)=EXP(PARAM(I,J))
15 CONTINUE
WRITE(6,950) J,(PARAM(I,J),I=NPX,1,-1)
WRITE(12,951) (PARAM(I,J),I=NPX,1,-1)
ENDIF
950 FORMAT(12,3X,7E12.4)
951 FORMAT(7E12.4)
10 CONTINUE
C
CLOSE(12)
RETURN
END
C
SUBROUTINE DATIN(X,XB,B,C,W,NPT,NPM,IFLAG,NPS)
C
C This subroutine reads the input data stored in file YOUNGS.PRM
C This file is from program TRAC2
C
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 X('),XB('),B('),W('),C(')
INTEGER NPS(')
C
WRITE(6,'') '>> Order of Regression Equation for Et?'
WRITE(6,'') (1 5) Polynomial 11:Power'
READ(5,'') NPM
WRITE(6,'') '>> Order of Regression Equation for Es?'
WRITE(6,'') (1 5) Polynomial 11:Power'
READ(5,'') NPM1
IFLAG=0
OPEN(15,FILE='youngs.prm')
READ(15,'') NPT
WRITE(6,910) NPT,NPM,NPM1
910 FORMAT(' Data Points= ',3, ' Order of Reg = ',23,')
C
DO 10 I=1,NPT
READ(15,'') X(I),B(I),C(I)
WRITE(6,'') X(I),B(I),C(I)
IF (NPM.EQ.11) THEN

```

```

XB(I)=X(I)
X(I)=DLOG(X(I))
B(I)=DLOG(B(I))
ENDIF
IF (NPM1.EQ.11) C(I)=DLOG(C(I))
10 CONTINUE
DO 12 I=1,NPT
W(I)=1.0D+0
12 CONTINUE
CLOSE(15)
NPS(1)=NPM
NPS(2)=NPM1
RETURN
END
C
SUBROUTINE GENER(A,W,B,NPT,NPX,AA,BB,NA,NB)
C
C This subroutine calculates the right-hand side of the
C simultaneous linear algebraic equations.
C
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 A(NA,'),W('),B('),AA(NB,'),BB('),AT(25,80)
C
IF (NPX.LE.10) THEN
NPM=NPX
ELSE
NPM=NPX-9
ENDIF
C
DO 10 I=1,NPM
DO 10 K=1,NPT
AT(I,K)=A(K,I)*W(K)
10 CONTINUE
C
DO 20 I=1,NPM
SUM=0.0D+0
DO 30 J=1,NPM
SUM1=0.0D+0
DO 35 K=1,NPT
SUM1=SUM1+AT(I,K)*A(K,J)
35 CONTINUE
AA(I,J)=SUM1
30 CONTINUE
C
DO 40 K=1,NPT
SUM=SUM+AT(I,K)*B(K)
40 CONTINUE
BB(I)=SUM
20 CONTINUE
RETURN
END
C

```



```
C.....
C      TRAC3
C      -- TRAC Step 3: Tire Model Calculation --
C      Hiroshi Nakashima
C      Transport Technology Research Laboratory
C      Carleton University
C
C      << Friction Calculation Included, No Inflation Process >>
C
C      Input Data File(1) -> 'gene3d.dat'
C      Input Data File(2) -> 'youngs.coal'
C      Output Data File(1) -> 'contct'.disp'
C      Output Data File(2) -> 'load*.dat'
C      Output Data File(3) -> 'trac_res.out'
C
C      (Ver. 1.0, June 10, 1997)
C.....
C      IMPLICIT REAL*8 (A-H,O-Z)
C      CHARACTER*40 ITITLE
C      PARAMETER(NPMAX=700,NEMAX=400,NNODE=8,NMMAX=4)
C      PARAMETER(IKV=350000,ILOAD=NPMAX*3,IPRSC=150)
C      COMMON /COORD/ X(NPMAX),Y(NPMAX),Z(NPMAX)
C      COMMON /ELNOD/ LNODE(NEMAX,NNODE)
C      COMMON /BOUND/ NBC(NPMAX,3)
C      COMMON /MATER/ NMAT(NEMAX),YOUNG(NMMAX),POISS(NMMAX)
C      COMMON /DISPL/ UX(NPMAX),UY(NPMAX),UZ(NPMAX)
C      COMMON /CONTC/ ICNT(NPMAX)
C      COMMON /PARAM/ TA(2,10),NT,NT1
C      COMMON /REACT/ FX(NPMAX),FY(NPMAX),FZ(NPMAX)
C      COMMON /MEINF/ NP,NE,NM,NB
C      COMMON /GEOMS/ WIDTH(100),CONTL(100),WLOAD(100),DEFLS(100)
C      COMMON /BUFFS/ GF(100),XG(100),VR(100)
C      REAL*8 BEE(6,24),BTDB(24,24),COORD(8,3),DEE(6,6)
C      REAL*8 DER(3,8),DERIV(3,8),DBEE(6,24),EKM(24,24)
C      REAL*8 ELD(24),SAMP(2,2)
C      REAL*8 TKV(IKV),TREAC(ILOAD),DREAC(ILOAD),DREAC1(ILOAD)
C      REAL*8 XJAC(3,3),XJAC1(3,3),SAMP1(2,2)
C      REAL*8 TKVIL(IPRSC,ILOAD),TKVIR(IPRSC)
C
C      INTEGER IG(NEMAX,24),KDIAG(ILOAD),ICNOD(IPRSC)
C      DATA ISAMP/2,IBTDB,IKM,BT,IDOF,4*24,IDEE,IBEE,IDBEE,14*6/
C      DATA ICOORD,NOD,2*8,IJAC,IJAC1,IDER,IDERIV,NDOOF,IT,6*3/
C      DATA IDER2/
C
C      Parameter Setup
C.....
C      open(6,FILE='error.msg')
C      ISTAGE=0
C      NGP=1
C      CALL GAUSS(SAMP,ISAMP,NGP)
C      CALL GAUSS(SAMP,ISAMP,NGP)
```

```

DO 45 I=1,NGP
DO 45 J=1,NGP
DO 45 K=1,NGP
-- Stiffness Matrix
CALL FMULN3(IDER,IDER,SAMP,ISAMP,I,J,K)
CALL MATMUL(DER,IDER,COORD,COORD,XJAC,UAC,IT,NOD,IT)
CALL TREE3(XJAC,UAC,XJAC1,UAC1,DET,IE)
CALL MATMUL(XJAC1,UAC1,DER,IDER,DERIV,IDERIV,IT,IT,NOD)
CALL NULL(BEE,IBEE,IH,IDO)
CALL FORMB3(BEE,IBEE,DERIV,IDERIV,NOD)
CALL MATMUL(DEE,IDE,IBEE,IBEE,DBEE,IBEE,IH,IDO)
QUOT=DET*SAMP(I,2)*SAMP(J,2)*SAMP(K,2)
CALL BTDBCC(BTD8,BEE,DBEE,IBTD8,IBEE,IBEE,QUOT)
CALL MATADD(EKM,IKM,BTD8,IBTD8,IDO,IDO)
45 CONTINUE
CALL FSPARV(TKV,EKM,IKM,IG,KDIAG,IDO,IE,NEMAX)
40 CONTINUE
C-----
C Right-hand side Arrangement for IStage>1
C-----
CALL LINMULS(TKV,TREAC,DREAC,NB,KDIAG)
CALL VEC SUB(DREAC,DREAC1,DREAC,NB)
C-----
C Prescribed Displacement Condition for IStage>1
C-----
DO 50 I=1,IPC
I1=ICNOD(I)
J=NBC(I,2)
CALL PRESCD(I,J,KDIAG,NB,TKV,DREAC,TKVIL,TKVIR,ILOAD,
* 0.0D+0,IPRESC)
50 CONTINUE
C-----
C Solve Linear Equations by Skyline Method
C-----
WRITE(6,*) 'SPARIN'
CALL SPARIN(TKV,NB,KDIAG)
WRITE(6,*) 'SPABAC'
CALL SPABAC(TKV,DREAC,NB,KDIAG)
C-----
C Check of Convergence
C-----
IEND=0
XX=0.0D+0
DO 55 I=1,NB
XX=XX+DREAC(I)*DREAC(I)
55 CONTINUE
VNORM1=DSQRT(XX)
RATIO=VNORM1/VNORM
WRITE(6,207) KI,RATIO
907 FORMAT(' -[Norm(1,2),:F8.4,]-----')
IF (RATIO .LT. 1.0D-3) IEND=1
IF (FRCFAC.EQ.0.0D+0) IEND=1

```

```

IF (ICONT(I) EQ 1) THEN
IPC=IPC+1
ICNOD(IPC)=I
J=NBC(I,2)
TREAC(I)=PRESCV
write(6,*) I,preascv
ENDIF
25 CONTINUE
C
WRITE(6,895) PRESCV
895 FORMAT(' [Prescribed Nodes] with Displ .',F8.4)
WRITE(6,962) (ICNOD(I) I=1,IPC)
962 FORMAT(3X,8I6)
IPENE=0
KI=0
C-----
C Newton-Raphson Friction Loop
C-----
700 CONTINUE
CALL NULVEC(TKV,NN)
KI=KI+1
WRITE(6,903) KI
903 FORMAT(' -[NR Loop,1,2,]-----')
C
CALL NULVEC(DREAC,NB)
C
IF (KI GT 1) THEN
DO 30 I=1,IPC
J2=NBC(ICNOD(I),2)
DREAC1(I,2)=0.0D+0
30 CONTINUE
ENDIF
XX=0.0D+0
DO 35 I=1,NB
XX=XX+TREAC(I)*TREAC(I)
35 CONTINUE
VNORM=DSQRT(XX)
C-----
C Element Stiffness Integration and Assembly
C IStage=1: Non-iterative solution
C IStage>1: Newton-Raphson iteration
C-----
WRITE(6,*) 'STIFF'
DO 40 IE=1,NE
CALL ECOORD(IE,COORD,COORD)
CALL NULL(EKM,IKM,IDO,IDO)
IEMAT=NMAT(IE)
CALL EVALU(EIEMAT,TVERT,E)
V=0.48D0
E=E/10.D0
CALL FORMD3(IDE,IDE,IE,V)
-- Gauss Integration Loop
C

```



```

C-----
C
C SUBROUTINE SAVGEM(ISTAGE)
C
C This subroutine saves the the contact geometry, such as deflections,
C contact lengths and contact widths, and normal reactions in file
C 'trac_res.out'.
C
C IMPLICIT REAL*8 (A-H,O-Z)
C COMMON GEOMS/ WIDTH(100),CONTL(100),WLOAD(100),DEFLS(100)
C
C OPEN(16,FILE='trac_res.out')
C WRITE(16,'') ISTAGE
C DO 10 I=1,ISTAGE
C   WRITE(16,900) DEFLS(I),WIDTH(I),CONTL(I),WLOAD(I)
C 10 CONTINUE
C 900 FORMAT(4F15.5)
C CLOSE(16)
C RETURN
C END
C-----
C SUBROUTINE EVALUE(IMAT,TDX,E)
C
C This subroutine interpolates the values of Young's moduli using
C polynomials obtained from TRAC2.
C
C IMPLICIT REAL*8 (A-H,O-Z)
C COMMON /PARAM/ TA(2,10),NT,NT1
C
C TDX : Vertical Load (KN)
C
C IF (IMAT.EQ.2) THEN
C   NTX=NT1
C   IF (NT1.EQ.1) NTX=NT1-9
C ELSE
C   NTX=NT
C   IF (NT.EQ.1) NTX=NT-9
C ENDIF
C
C E=TA(IMAT,NTX)
C
C IF (((IMAT.EQ.1) .AND. (NT.EQ.1)) .OR.
C   ((IMAT.EQ.2) .AND. (NT1.EQ.1))) THEN
C   IF (TDX.LE.1.0) THEN
C     E=E*1.0D0**TA(IMAT,1)
C   ELSE
C     E=E*TDX**TA(IMAT,1)
C   ENDIF
C ELSEIF ((NT.LE.10) .OR. (NT1.LE.10)) THEN
C   DO 10 I=2,NTX
C     J=NTX-I+1
C     E=E*TA(IMAT,J)*TDX**(I-1)

```

```

10 CONTINUE
ENDIF
RETURN
END
C-----
C SUBROUTINE PARAMS(YLDMAX)
C IMPLICIT REAL*8 (A-H,O-Z)
C COMMON /PARAM/ TA(2,10),NT,NT1
C
C OPEN(18,FILE='Youngs.coef')
C REA,Y(18,'') NT
C KN=NT
C IF (NT.EQ.1) KN=NT-9
C READ(18,'') (TA(1,K),K=1,KN)
C READ(18,'') NT1
C KN=NT1
C IF (NT1.EQ.1) KN=NT1-9
C READ(18,'') (TA(2,K),K=1,KN)
C READ(18,'') YLDMAX
C CLOSE(18)
C RETURN
C END

```

```

C-----
C
C THIS BLOCK OF CODES IS USED BY PROGRAMS trac1, trac2 AND trac3
C-----
C
SUBROUTINE BTDBCC(BTD8,BEE,DBEE,IBTD8,IBEE,DBEE,QUOT)
C-----
C
C This subroutine calculates the product,
C
C [B]T [D] [B]
C
C IMPLICIT REAL*8 (A-H,O-Z)
C REAL*8 BTD8(1BTD8*),BEE(1BEE*),DBEE(1DBEE*),
C
C DO 10 I=1,24
C DO 10 J=1,24
C X=0.0D+0
C DO 15 K=1,6
C X=X+BEE(X,I)*DBEE(K,J)
C 15 CONTINUE
C BTDB(I,J)=X*QUOT
C 10 CONTINUE
C RETURN
C END
C-----
C
SUBROUTINE LINMLS(BP,DISPS,TLOAD,N,KDIAG)
C-----
C
C This subroutine processes the force vector, and obtains the
C total vertical reaction.
C
C IMPLICIT REAL*8 (A-H,O-Z)
C REAL*8 BP(*),TLOAD(*),DISPS(*)
C INTEGER KDIAG(*)
C
C DO 2 I=1,N
C X=0.0D+0
C LUP=KDIAG(I)
C IF (I.EQ.1) LOW=LUP
C IF (I.NE.1) LOW=KDIAG(I-1)+1
C DO 3 J=LOW,LUP
C X=X+BP(J)*DISPS(I-J-LUP)
C 3 CONTINUE
C TLOAD(I)=X
C IF (I.EQ.1) GOTO 2
C LUP=LUP-1
C DO 4 J=LOW,LUP
C K=I-J-LUP-1
C TLOAD(K)=TLOAD(K)+BP(J)*DISPS(I)
C 4 CONTINUE
C 2 CONTINUE
C RETURN

```

```

C-----
C
SUBROUTINE EMSSGE(I,N,N1)
C-----
C
C This subroutine prints out some error messages.
C
C GOTO (1,2,3) I
C 1 WRITE(6,900) N,N1
C 900 FORMAT('TRAC ERROR: KDIAG(NB)=',I6,' > IKV=',I6)
C GOTO 100
C 2 WRITE(6,910) N,N1
C 910 FORMAT('TRAC ERROR: NR-ITERATION=',I3,' > ',I3)
C GOTO 100
C 3 GOTO 100
C 100 STOP
C RETURN
C END
C-----
C
SUBROUTINE SETBC
C-----
C
C This subroutine finds the total degrees of freedom of the system,
C and eliminates all constrained nodes.
C
C IMPLICIT REAL*8 (A-H,O-Z)
C PARAMETER(NPMAX=700)
C COMMON /BOUND/ NBC(NPMAX,3)
C COMMON /MEINF/ NP,NE,NM,NB
C
C Calculation for Total Degrees of Freedom
C L=0
C DO 10 I=1,NP
C DO 10 K=1,3
C IF (NBC(I,K).EQ.1) THEN
C NBC(I,K)=0
C ELSE
C L=L+1
C NBC(I,K)=L
C ENDIF
C 10 CONTINUE
C NB=L
C RETURN
C END
C-----
C
SUBROUTINE ECOORD(IE,COORD,ICCOORD)
C-----
C
C This subroutine assigns coordinates from X, Y, Z to COORD
C
C IMPLICIT REAL*8 (A-H,O-Z)
C PARAMETER(NPMAX=700,NEMAX=400,NNODE=8)
C COMMON /COORD3/ X(NPMAX),Y(NPMAX),Z(NPMAX)
C COMMON /ELNOD/ LNODE(NEMAX,NNODE)
C REAL*8 COORD(ICOORD,*)

```



```

C      DO 20 I=1,8
C      COORD(1,1)=X(LNODE(IE,I))
C      COORD(1,2)=Y(LNODE(IE,I))
C      COORD(1,3)=Z(LNODE(IE,I))
20 CONTINUE
RETURN
END

C-----
C      SUBROUTINE IGSET(IE,IG)
C
C      This subroutine assigns the degrees of freedom stored in NBC to IG.
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      PARAMETER(NPMAX=700,NEMAX=400,NNODE=8)
C      COMMON /ELNOD/ LNODE(NEMAX,NNODE)
C      COMMON /BOUND/ NBC(NPMAX,3)
C      INTEGER IG(NEMAX,*)
C
C      INC=0
C      DO 10 I=1,8
C      DO 10 J=1,3
C      INC=INC+1
C      IG(IE,INC)=NBC(LNODE(IE,I),J)
10 CONTINUE
RETURN
END

C-----
C      SUBROUTINE FORMD3(DEE,IDE,E,V)
C
C      calculate the elastic property matrix [D]
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      REAL*8 DEE(IDE,*)
C
C      TEMP1=E*(1.D0-V)/((1.D0+V)*(1.D0-2.D0*V))
C      TEMP2=V/(1.D0-V)
C      TEMP3=0.5D0*(1.D0-2.D0*V)/(1.D0-V)
C      DO 1 I=1,IDE
C      DO 1 J=1,IDE
C      DEE(I,J)=0.D0+0
1 CONTINUE
DEE(1,1)=TEMP1
DEE(2,2)=TEMP1
DEE(3,3)=TEMP1
DEE(1,2)=TEMP1*TEMP2
DEE(2,1)=DEE(1,2)
DEE(1,3)=DEE(1,2)
DEE(3,1)=DEE(1,3)
DEE(2,3)=DEE(1,3)
DEE(3,2)=DEE(2,3)
DEE(4,4)=TEMP1*TEMP3

```

```

DEE(5,5)=DEE(4,4)
DEE(6,6)=DEE(5,5)
RETURN
END

C-----
C      SUBROUTINE FORMB3(BEE,IBEE,DERIV,IDERIV,NOD)
C
C      [B] matrix in FEM formulation.
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      REAL*8 BEE(IBEE,*),DERIV(IDERIV,*)
C
C      DO 1 M=1,NOD
C      N=3*M
C      K=N-1
C      L=K-1
C      BEE(1,L)=DERIV(1,M)
C      BEE(4,K)=DERIV(1,M)
C      BEE(6,N)=DERIV(1,M)
C      BEE(2,K)=DERIV(2,M)
C      BEE(4,L)=DERIV(2,M)
C      BEE(5,N)=DERIV(2,M)
C      BEE(3,N)=DERIV(3,M)
C      BEE(6,L)=DERIV(3,M)
1 CONTINUE
RETURN
END

C-----
C      SUBROUTINE GAUSS(SAMP,ISAMP,NGP)
C
C      This subroutine assigns the Gauss integration constants.
C
C      REAL*8 SAMP(ISAMP,*)
C
C      GOTO (1,2) NGP
1 SAMP(1,1)=0.D0
SAMP(1,2)=2.D0
GOTO 100
2 SAMP(1,1)=1.D0/DSQRT(3.D0+0)
SAMP(2,1)=SAMP(1,1)
SAMP(1,2)=1.D0
SAMP(2,2)=1.D0
100 RETURN
END

C-----
C      SUBROUTINE NULVEC(VEC,N)
C
C      This subroutine initializes a vector (A) with zeroes.
C
C      REAL*8 VEC(*)

```

```

DO 1 I=1,N
  VEC(I)=0.0D+0
1 CONTINUE
RETURN
END
C-----
SUBROUTINE NULL(A,IA,M,N)
C
C This subroutine initializes a M by N matrix [A] with zeroes.
C
REAL*8 A(IA,*)
DO 1 I=1,M
DO 1 J=1,N
  A(I,J)=0.0D+0
1 CONTINUE
RETURN
END
C-----
SUBROUTINE MVMULT(XM,IMV,K,L,Y)
C
C This subroutine calculates the product of a vector and
C a matrix  $Y(I)=\sum_{J=1}^L XM(I,J)*V(J)$ 
C
REAL*8 XM(IM,*,),V(*),Y(*)
DO 1 I=1,K
  Y(I)=0.0D+0
DO 2 J=1,L
  Y(I)=Y(I)+XM(I,J)*V(J)
2 CONTINUE
1 CONTINUE
RETURN
END
C-----
SUBROUTINE MTVMUL(XM,IMV,K,L,Y)
C
C This subroutine calculates the product of a vector and
C a matrix  $Y(I)=\sum_{J=1}^L XM(J,I)*V(J)$ 
C
REAL*8 XM(IM,*,),V(*),Y(*)
DO 1 I=1,L
  Y(I)=0.0D+0
DO 2 J=1,K
  Y(I)=Y(I)+XM(J,I)*V(J)
2 CONTINUE
1 CONTINUE
RETURN
END
C-----
SUBROUTINE MATMUL(A,IA,IB,IC,IA,M,N)
C
C This subroutine performs matrix multiplication  $[C]=A[B]$ 
C
REAL*8 A(IA,*),B(IB,*,),C(IC,*)X
DO 1 I=1,L
DO 1 J=1,N
  X=0.0D+0
DO 2 K=1,M
  X=X+A(I,K)*B(K,J)
2 CONTINUE
C(I,J)=X
1 CONTINUE
RETURN
END
C-----
SUBROUTINE FMIN3(IDER,IDER,SAMP,SAMP,I,J,K)
C
C This subroutine calculates gradient matrix DER,
C derivatives of shapge functions.
C
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 DER(IDER,*,),SAMP(SAMP,*)
ETA =SAMP(K,1)
XI =SAMP(J,1)
ZETA=SAMP(1,1)
ETAM =1.D0-ETA
XIM =1.D0-XI
ZETAM=1.D0-ZETA
ETAP =1.D0+ETA
XIP =1.D0+XI
ZETAP=1.D0+ZETA
C
C -- Derivative of gza
DER(1,1)=-0.125D0*ETAM*ZETAP
DER(1,2)=-0.125D0*ETAP*ZETAP
DER(1,3)=-0.125D0*ETAP*ZETAP
DER(1,4)=-0.125D0*ETAM*ZETAP
DER(1,5)=-0.125D0*ETAM*ZETAM
DER(1,6)=-0.125D0*ETAP*ZETAM
DER(1,7)=-0.125D0*ETAP*ZETAM
DER(1,8)=-0.125D0*ETAM*ZETAM
C
C -- Derivative of eta
DER(2,1)=-0.125D0*XIM*ZETAP
DER(2,2)=-0.125D0*XIM*ZETAP
DER(2,3)=-0.125D0*XIP*ZETAP
DER(2,4)=-0.125D0*XIP*ZETAP
DER(2,5)=-0.125D0*XIM*ZETAM
DER(2,6)=-0.125D0*XIM*ZETAM
DER(2,7)=-0.125D0*XIP*ZETAM
DER(2,8)=-0.125D0*XIP*ZETAM
C
C -- Derivative of zeta

```

```

C
DER(3,1)= 0.125D0*XIM*ETAM
DER(3,2)= 0.125D0*XIM*ETAP
DER(3,3)= 0.125D0*XIP*ETAP
DER(3,4)= 0.125D0*XIP*ETAM
DER(3,5)= 0.125D0*XIM*ETAM
DER(3,6)= 0.125D0*XIM*ETAP
DER(3,7)= 0.125D0*XIP*ETAP
DER(3,8)= 0.125D0*XIP*ETAM
RETURN
END
C-----
SUBROUTINE FMLIN2(IDER2,IDER2,FUN2,SAMP,ISAMP,J,K)
C
C This subroutine calculates 2D shape functions and derivatives
C
C
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 DER2(IDER2,:),FUN2(:),SAMP(ISAMP,:)
C
ETA =SAMP(K,1)
XI =SAMP(J,1)
ETAM=1.D0-ETA
XIM =1.D0-XI
ETAP=ETA +1.D0
XIP =XI +1.D0
FUN2(1)= 0.25D0*XIM*ETAM
FUN2(2)= 0.25D0*XIM*ETAP
FUN2(3)= 0.25D0*XIP*ETAP
FUN2(4)= 0.25D0*XIP*ETAM
DER2(1,1)= 0.25D0*ETAM
DER2(1,2)= 0.25D0*ETAP
DER2(1,3)= 0.25D0*ETAP
DER2(1,4)= 0.25D0*ETAM
DER2(2,1)= 0.25D0*XIM
DER2(2,2)= 0.25D0*XIM
DER2(2,3)= 0.25D0*XIP
DER2(2,4)= 0.25D0*XIP
RETURN
END
C-----
SUBROUTINE DET2D(IDER2,IDER2,COORD,ICOORD,AIRP,PRESS)
C
C This subroutine calculates
C the Jacobian determinant |J| for 2D.
C
REAL*8 DER2(IDER2,:),COORD(ICOORD,:),AIRP(*)
REAL*8 DEFFR(2,2)
C
DO 10 I=1,3
DO 20 J=1,2
DO 30 K=1,2
DEFFR(J,K)=0.D+0
20 CONTINUE
C
C-----
SUBROUTINE AIRPRS(AIRP,EAIR,FUN2)
C
C This subroutine interpolates the inflation pressure.
C
REAL*8 AIRP(*),EAIR(*),FUN2(*)
C
DO 10 I=1,4
DO 20 J=1,3
L=3*(I-1)+J
EAIR(L)=FUN2(I)*AIRP(J)
20 CONTINUE
10 CONTINUE
RETURN
END
C-----
SUBROUTINE TREEK3(XJAC,UJAC,XJAC1,UJAC1,DET,IE)
C
C This subroutine calculates
C the Jacobian determinant |J| and the inverse of |J|
C
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 XJAC(UJAC,:),XJAC1(UJAC1,:)
C
DET= XJAC(1,1)*XJAC(2,2)*XJAC(3,3)-XJAC(3,2)*XJAC(2,3))
DET=DET-XJAC(1,2)*XJAC(2,1)*XJAC(3,3)+XJAC(3,1)*XJAC(2,3))
DET=DET+XJAC(1,3)*XJAC(2,1)*XJAC(3,2)-XJAC(3,1)*XJAC(2,2))
IF (DET.LT.1.0D-30) THEN
WRITE(6,900) DET,IE
900 FORMAT(/,'TIDRAC ERROR: Jacobian 'E15.7,' <= 0 at ',I4)
STOP
ENDIF
XJAC1(1,1)=(-XJAC(2,2)*XJAC(3,3)-XJAC(3,2)*XJAC(2,3))/DET
XJAC1(2,1)=(-XJAC(2,1)*XJAC(3,3)+XJAC(3,1)*XJAC(2,3))/DET
XJAC1(3,1)=(-XJAC(2,1)*XJAC(3,2)+XJAC(3,1)*XJAC(2,2))/DET
XJAC1(1,2)=(-XJAC(1,2)*XJAC(3,3)+XJAC(3,2)*XJAC(1,3))/DET
XJAC1(2,2)=(-XJAC(1,1)*XJAC(3,3)-XJAC(3,1)*XJAC(1,3))/DET

```

```

XJAC(3,2)=(-XJAC(1,1)*XJAC(3,2)+XJAC(3,1)*XJAC(1,2))/DET
XJAC(1,1)=XJAC(1,2)*XJAC(2,3)-XJAC(2,2)*XJAC(1,3)/DET
XJAC(2,3)=(-XJAC(1,1)*XJAC(2,3)+XJAC(2,1)*XJAC(1,3))/DET
XJAC(3,3)=(-XJAC(1,1)*XJAC(2,2)+XJAC(2,1)*XJAC(1,2))/DET
RETURN
END

```

```

C-----
SUBROUTINE MATRAN(A,IA,IB,M,N)

```

```

C
C This subroutine performs the inverse of a matrix.

```

```

C
C REAL*8 A(IA,*)B(IB,*)

```

```

C
C DO 1 I=1,M

```

```

C DO 1 J=1,N

```

```

C A(JI)=B(I,J)

```

```

C 1 CONTINUE

```

```

C RETURN

```

```

C END

```

```

C-----
SUBROUTINE MSMULT(A,IA,C,M,N)

```

```

C
C This subroutine calculates the product of a matrix and a constant.

```

```

C
C REAL*8 A(IA,*)C

```

```

C DO 1 I=1,M

```

```

C DO 1 J=1,N

```

```

C A(IJ)=A(I,J)*C

```

```

C 1 CONTINUE

```

```

C RETURN

```

```

C END

```

```

C-----
SUBROUTINE VSMULT(A,C,M)

```

```

C
C This subroutine calculates the product of a vector and a constant

```

```

C
C REAL*8 A(*)C

```

```

C DO 1 I=1,M

```

```

C A(I)=A(I)*C

```

```

C 1 CONTINUE

```

```

C RETURN

```

```

C END

```

```

C-----
SUBROUTINE MATADD(A,IA,B,IB,M,N)

```

```

C
C This subroutine performs a matrix addition

```

```

C
C REAL*8 A(IA,*)B(IB,*)

```

```

C DO 1 I=1,M

```

```

DO 1 J=1,N
A(I,J)=A(I,J)+B(I,J)
1 CONTINUE
RETURN
END

```

```

C-----
SUBROUTINE VECADD(A,B,C,M)

```

```

C
C This subroutine performs a vector addition.

```

```

C
C REAL*8 A(*)B(*)C(*)

```

```

C DO 1 I=1,M

```

```

C A(I)=B(I)+C(I)

```

```

C 1 CONTINUE

```

```

C RETURN

```

```

C END

```

```

C-----
SUBROUTINE VECSUB(A,B,C,M)

```

```

C
C This subroutine performs a vector subtraction.

```

```

C
C REAL*8 A(*)B(*)C(*)

```

```

C DO 1 I=1,M

```

```

C A(I)=B(I)-C(I)

```

```

C 1 CONTINUE

```

```

C RETURN

```

```

C END

```

```

C-----
SUBROUTINE SPARIN(A,N,KL G)

```

```

C
C Gaussian elimination using Sky-line method

```

```

C
C IMPLICIT REAL*8 (A-H,O-Z)

```

```

C REAL*8 A(*)

```

```

C INTEGER KDIAG(*)

```

```

C A(1)=DSORT(A(1))

```

```

C DO 1 I=2,N

```

```

C KI=KDIAG(I-1)

```

```

C L=KDIAG(I-1)+KI+1

```

```

C DO 2 J=L,I

```

```

C X=A(KI+J)

```

```

C KJ=KDIAG(J)-J

```

```

C IF (J.EQ.1) GOTO 2

```

```

C LBAR=KDIAG(J-1)+KJ+1

```

```

C LBAR=MAXX(LBAR)

```

```

C IF (LBAR.EQ.J) GOTO 2

```

```

C M=J-1

```

```

C DO 3 K=LBAR,M

```

```

C X=X-A(KI+K)*A(KJ+K)

```

```

3      CONTINUE
2      A(K1+J)=X/A(K1+J)
      A(K1+J)=DSORT(X)
1      CONTINUE
      RETURN
      END
C-----
SUBROUTINE SPABAC(A,B,N,KDIAG)
C
C      Back substitution
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(*),B(*)
      INTEGER KDIAG(*)
C
      B(1)=B(1)/A(1)
      DO 1 I=2,N
        KI=KDIAG(I)-1
        L=KDIAG(I-1)-KI+1
        X=B(I)
        IF (L.EQ.1) GOTO 1
        M=I-1
        DO 2 J=L,M
          X=X-A(K1+J)*B(J)
2      CONTINUE
1      B(I)=X/A(K1+1)
      DO 3 IT=2,N
        I=N+2-IT
        KI=KDIAG(I)-1
        X=B(I)/A(K1+1)
        B(I)=X
        L=KDIAG(I-1)-KI+1
        IF (L.EQ.1) GOTO 3
        M=I-1
        DO 4 K=L,M
          B(K)=B(K)-X*A(K1+K)
4      CONTINUE
3      CONTINUE
      B(1)=B(1)/A(1)
      RETURN
      END
C-----
SUBROUTINE FSPARV(TKV,EKM,IKM,IG,KDIAG,IDO,I,NEMAX)
C
C      This subroutine assembles global stiffness matrix.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 TKV(*),EKM(IKM,*),
      INTEGER KDIAG(*),IG(NEMAX,*)
C
      DO 1 I=1,IDO
        K=IG(I)

```

```

      IF (K.EQ.0) GOTO 1
      DO 2 J=1,IDO
        IF (IG(I,J).EQ.0) GOTO 2
        IW=K-IG(I,J)
        IF (IW.LT.0) GOTO 2
        IVAL=KDIAG(K)-IW
        TKV(IVAL)=TKV(IVAL)+EKM(I,J)
2      CONTINUE
1      CONTINUE
      RETURN
      END
C-----
SUBROUTINE FSPAR1(TV,EK,IG,IDO,I,NEMAX)
C
C      This subroutine is never used.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 TV(*),EK(*)
      INTEGER IG(NEMAX,*)
C
      DO 1 I=1,IDO-12
        K=IG(I+12)
        IF (K.EQ.0) GOTO 1
        TV(K)=TV(K)+EK(I)
1      CONTINUE
      RETURN
      END
C-----
SUBROUTINE FKDIAG(KDIAG,IG,IDO,I,NEMAX)
C
C      This subroutine determines the size of global stiffness matrix
C
      INTEGER KDIAG(*),IG(NEMAX,*)
C
      DO 1 I=1,IDO
        IWP1=1
        IF (IG(I),EQ.0) GOTO 1
        DO 2 J=1,IDO
          IF (IG(I,J).EQ.0) GOTO 2
          IM=IG(I,J)-IG(I,J)+1
          IF (IM.GT.IWP1) IWP1=IM
2      CONTINUE
        K=IG(I)
        IF (IWP1.GT.KDIAG(K)) KDIAG(K)=IWP1
1      CONTINUE
      RETURN
      END
C-----
SUBROUTINE PRESCD(I,L,KDIAG,NB,TKV,TREAC,TKVIL,TKVIR,IS,
      PRESCV,IQ)
C
C      This subroutine processes the nodes with prescribed displacements.

```

```

C      IMPLICIT REAL*8 (A-H,O-Z)
C      INTEGER KDIAG(I)
C      REAL*8 TKV(I),TKVIL(IQ,I),TREAC(I),TKVIR(I)

C      -- Assigned Large Number
C      FACTOR=1.0D+18
C      DO 5 K=1,NB
C        TKVIL(I,K)=0.0D+0
C      5 CONTINUE
C      -- Left Hand Side
C      IF (L.EQ.1) THEN
C        K=KDIAG(L)
C        J1=K
C        TKVIL(J1)=TKV(K)
C        TKV(K)=FACTOR*TKV(K)
C      ELSE
C        K=KDIAG(L)-KDIAG(L-1)
C        DO 10 J=K,1,-1
C          J1=KDIAG(L)-J+1
C          J2=L-J+1
C          TKVIL(J2)=TKV(J1)
C          IF (J.EQ.1) TKV(KDIAG(L))=FACTOR*TKV(KDIAG(L))
C      10 CONTINUE
C      ENDIF
C      N=0
C      DO 15 K=L+1,NB
C        K1=KDIAG(K)-KDIAG(K-1)
C        N=N+1
C        IF (K1.EQ.N) GOTO 15
C        TKVIL(L+1)=TKV(KDIAG(K)-N)
C      15 CONTINUE
C      -- Right Hand Side . TKV=Factor*TKV
C      TKVIR(I)=TREAC(L)
C      TREAC(L)=PRESV*TKV(KDIAG(L))
C      RETURN
C      END

C      SUBROUTINE SEARCH(ICNOD,IPRESV,PRESV,ISER,DEFMAX)
C      C      This subroutine searches for the probable contact points
C      C      IMPLICIT REAL*8 (A-H,O-Z)
C      C      PARAMETER(NPMA=700)
C      C      COMMON /COORD/ X(NPMA),Y(NPMA),Z(NPMA)
C      C      COMMON /MEINF/ NP,NE,NM,NB
C      C      COMMON /CONTC/ ICNT(NPMA)
C      C      INTEGER ICNOD(IPRESV)
C      C      DLM=DEFMAX*150.D0

C      WRITE(' ','SEARCH')
C      ISER=0
C      YC=Y(ICNT(1))
C      DMINO=0.0D+0
C      100 CONTINUE
C      DMIN=1.0D+10
C      DO 10 I=1,NP
C        IF (ICNT(I).EQ.1) GOTO 10
C        DY=DABS(YC-Y(I))
C        IF (DY.LT.DMIN) THEN
C          DMIN=DY
C          JNEXT=I
C        ENDIF
C      10 CONTINUE
C      PRESVO=DMIN+DMINO
C      PRESV=PRESVO*1.5D0
C      IF (PRESV.GT.2.D0*DLIM) PRESV=2.D0*DLIM
C      WRITE(6,920) JNEXT,PRESV
C      920 FORMAT(' --Next Probable Contact Node= ',I3,
C      * ' with displacement= ',F8.4)
C      ICNT(JNEXT)=1
C      C      -- Contact Condition <= DEFMAX/100.0 cm --
C      IF (PRESV.LE.DLIM) THEN
C        WRITE(6,900) JNEXT
C        900 FORMAT(' --Contact Detected at ',I3)
C        ICNT(JNEXT)=1
C        ISER=1
C        WRITE(6,910)
C        910 FORMAT(' --New Search for Next Contact')
C        DMINO=PRESV
C        GOTO 100
C      ENDIF
C      RETURN
C      END

C      SUBROUTINE PENETR(ICNOD,IPRESV,IPENE,PRESV,TREAC)
C      C      This subroutine finds the penetration points
C      C      IMPLICIT REAL*8 (A-H,O-Z)
C      C      PARAMETER(NPMA=700)
C      C      COMMON /COORD/ X(NPMA),Y(NPMA),Z(NPMA)
C      C      COMMON /DISPL/ UX(NPMA),UY(NPMA),UZ(NPMA)
C      C      COMMON /BOUND/ NBC(NPMA,3)
C      C      COMMON /MEINF/ NP,NE,NM,NB
C      C      COMMON /CONTC/ ICNT(NPMA)
C      C      REAL*8 TREAC(I)
C      C      INTEGER ICNOD(IPRESV)
C      C      WRITE(' ','PENETR')

```

```

      Y0=Y(ICNOD(I))
      YC=Y0+TREAQ(NBC(ICNOD(I),2))
      DO 10 I=1,NP
      IF (ICONT(I).GE.0) GOTO 10
      IF (IEQ(ICNOD(I))) GOTO 10
      J1=NBC(I,2)
      YY=Y(I)+TREAQ(J1)
      DY=YY-YC
      C
      C      -- Detect Penetrated Part of Nodes
      IF (DY LT -.075D0) THEN
        DY1=DABS(DY)+DABS(UY(I))
        PRESQV=DABS(DY/DY1)*PRESQV/2.D0
        WRITE(6,901) I,PRESQV
      901  FORMAT(' Penetrate at Node',I4,' --> Disp Modified to',F8.4)
      IPENE=1
      RETURN
      ENDIF
      10 CONTINUE
      IPENE=0
      RETURN
      END
      C-----
      SUBROUTINE UPDATE
      C
      C      This subroutine updates the coordinates of the system
      C
      C
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(NPMAX=700)
      COMMON /COORDS/ X(NPMAX), Y(NPMAX), Z(NPMAX)
      COMMON /DISPL/ UX(NPMAX), UY(NPMAX), UZ(NPMAX)
      COMMON /MEINF/ NP,NE,NM,NB
      C
      DO 10 I=1,NP
      X(I)=X(I)+UX(I)
      Y(I)=Y(I)+UY(I)
      Z(I)=Z(I)+UZ(I)
      10 CONTINUE
      RETURN
      END
      C-----
      SUBROUTINE UPDISP(NB,TREAQ,DREAQ,KI)
      C
      C      This subroutine updates the coordinates of the system.
      C
      C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 TREAQ(*),DREAQ(*)
      C
      BETA=1.D0+0
      DO 10 I=1,NB
      TREAQ(I)=TREAQ(I)+DREAQ(I)*BETA
      10 CONTINUE

```

```

      RETURN
      END
      C-----
      SUBROUTINE LDSAVE(ISTAGE,ICNOD,IPC,PRESQV,TFY)
      C
      C      This subroutine saves the load-deflection data in appropriate files
      C
      C
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(NPMAX=700,NEMAX=400)
      COMMON /REACT/ FX(NPMAX),FY(NPMAX),FZ(NPMAX)
      CHARACTER*80 IDFILE
      INTEGER ICNOD(*)
      C
      WRITE(6,903)
      903 FORMAT(' REACTION Result in (N)')
      IF (INT(ISTAGE/10) GE.1) THEN
        WRITE(IDFILE,96) ISTAGE
      96  FORMAT('load-',I2,'rs.dat')
      ELSE
        WRITE(IDFILE,99) ISTAGE
      99  FORMAT('load-',I1,'rs.dat')
      ENDIF
      OPEN(18,FILE=IDFILE)
      WRITE(18,'') IPC
      TFX=0.D0+0
      TFY=0.D0+0
      TFZ=0.D0+0
      DO 10 I=1,IPC
      J=ICNOD(I)
      TFX=TFX+FX(J)
      TFY=TFY+FY(J)
      TFZ=TFZ+FZ(J)
      WRITE(6,905) J,FX(J),FY(J),FZ(J)
      905  FORMAT(5X,I4,3F12.3)
      WRITE(18,900) J,FY(J),PRESQV
      900  FORMAT(13,2F15.7)
      10 CONTINUE
      WRITE(6,906) TFX,TFY,TFZ
      906 FORMAT(' Total',3F12.3)
      CLOSE(18)
      RETURN
      END
      C-----
      SUBROUTINE DATIN(ITLE,PRESS,FRCFAC,IP1,DEFMAX)
      C
      C      This subroutine reads input data, ie, FEM mesh
      C      and material properties.
      C
      C
      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*40 ITITLE
      PARAMETER(NPMAX=700,NEMAX=400,NNODE=8,NMMAX=4)
      COMMON /COORDS/ X(NPMAX), Y(NPMAX), Z(NPMAX)

```

```

COMMON /ELNOD/ : : IODE(NEMAX,NNODE)
COMMON /COORD/ : : X(NMAX),Y(NMAX),Z(NMAX)
COMMON /MATER/ NMAT(NEMAX),YOUNG(NMAX),POISS(NMAX)
COMMON /CONTC/ ICNT(NPMAX)
COMMON /MEINF/ NP,NE,NM,NB

C
OPEN(9,FILE='gene3d.dat')
READ(9, '(A40)') ITITLE
READ(9, *) PRESS,FRCFAC

C
C      Pressure Unit from kPa to N/cm^2
PRESS=PRESS/10.0
READ(9, *) NP,NE,IP1

C
DO 10 I=1,NP
  ICNT(I)=0
10 CONTINUE

C
READ(9, *) NMAT(1),Y(1),Z(1),(NBC(I,K),K=1,3)
IF (I.EQ.IP1) ICNT(I)=1
20 CONTINUE
DO 20 I=1,NE
  READ(9, *) NDUMMY,(LNODE(I,K),K=1,8),NMAT(I)
  READ(9, *) NM
  DO 30 I=1,NM
    READ(9, *) YOUNG(I),POISS(I)
    YOUNG(I)=YOUNG(I)/10.0
30 CONTINUE
CLOSE(9)

C
DEFMAX=-10000.00
DEFMIN= 10000.00
DO 40 I=1,NP
  IF (Y(I) GT DEFMAX) DEFMAX=Y(I)
  IF (Y(I) LT DEFMIN) DEFMIN=Y(I)
40 CONTINUE
DEFMAX=(DEFMAX+DEFMIN)*0.100

C
RETURN
END

C-----
SUBROUTINE DSAVE(ISTAGE,ITITLE)
C
C This subroutine saves the deflection geometry for each iteration
C
C
IMPLICIT REAL*8 (A-H,O-Z)
CHARACTER*40 ITITLE,IDFILE
PARAMETER(NPMAX=700,NEMAX=400,NNODE=8,NMMAX=4)
COMMON /COORD/ X(NPMAX),Y(NPMAX),Z(NPMAX)
COMMON /DISPL/ UX(NPMAX),UY(NPMAX),UZ(NPMAX)
COMMON /MEINF/ NP,NE,NM,NB

C
IF (INT(ISTAGE/10).GE.1) THEN
  WRITE(IDFILE,900) ISTAGE
900 FORMAT('cont',I2,'rs,disp')
ELSE
  WRITE(IDFILE,901) ISTAGE
901 FORMAT('contact',I1,'rs,disp')
ENDIF

C
OPEN(15,FILE=IDFILE)
WRITE(15, '(A40)') ITITLE
WRITE(15, *) NP,NE
WRITE(15,910) (K,X(K),Y(K),Z(K),UX(K),UY(K),UZ(K),K=1,NP)
910 FORMAT(15,6F12.5)
CLOSE(15)
RETURN
END

C-----
SUBROUTINE CONDAT(ISTAGE)
C
C This subroutine finds out the contact length, width and area
C (approximated by multiplying the length by the width),
C and writes them to the screen.
C
IMPLICIT REAL*8 (A-H,O-Z)
PARAMETER(NPMAX=700,NEMAX=400)
COMMON /COORD/ X(NPMAX),Y(NPMAX),Z(NPMAX)
COMMON /CONTC/ ICNT(NPMAX)
COMMON /MEINF/ NP,NE,NM,NB
COMMON /GEOMS/ WIDTH(120),CONTL(120),WLOAD(120)

C
XCMAX=X(1)
ZCMAX=Z(1)
DO 10 J=2,NP
  IF (ICNT(J).EQ.1) THEN
    IF (X(J) GT XCMAX) XCMAX=X(J)
    IF (Z(J) GT ZCMAX) ZCMAX=Z(J)
  ENDIF
10 CONTINUE
WIDTH(ISTAGE)=2.0*ZCMAX
CONTL(ISTAGE)=2.0*XCMAX
AREAS=4.0*ZCMAX*XCMAX
WRITE(6,901) WIDTH(ISTAGE),CONTL(ISTAGE),AREAS
901 FORMAT('Contact Width=',F8.4,' Length=',F8.4,' Area=',F8.4)
RETURN
END

```


Appendix C

Computer Source Codes for Simulation of Tire-Soft Terrain Interaction — Practical Model

Program CWVPM5 is the practical tire-terrain interaction model presented in this thesis.

```

C*****
C CWP/MS: Carleton Wheeled Vehicle Performance Model
C (Version 5.2, June 28, 1993)
C by Songtao Chen of
C Transport Technology Research Laboratory
C Department of Mechanical and Aerospace Engineering
C Carleton University
C
C Input data file(1) --> 'test.in'
C Output data file(1) --> 'test.out'
C Output data file(2) --> 'stress.out'
C*****
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C COMMON /BLOCKA/ PIR,XL,ZO,WIDTH,THETA0,THETA1,THETA2,
& STHETA,CTHETA,GAMMA0,GAMMA1,GAMMA2,SGAMMA
C
C DOUBLE PRECISION KC,KPHI,NN,K,K0,KU
C COMMON /BLOCKB/ KC,KPHI,NN,BB,K,K0,AU,ZU,KU,PU,SLIP,DJEQV,
& ZSINK,TANPHI,PK,A7,B7,PK2
C
C COMMON /BLOCKC/ RLLG,RCAP,HTLUG,GAMAOL,GAMAOC,GAMA1L,GAMA1C,
& GAMA2L,GAMA2C,THETOL,THETOC,THET1L,THET1C,
& THET2L,THET2C,RATIO,
& CRUB,PHIRUB,PKRUB,ARUB,BRUB,PK2RUB,TANFIL,
& CINT,PHINT,PKINT,AINT,BINT,PK2INT,TANFIC,
& ZOMAXL,ZOMAXC,DJEQVL,DJEQVC
C
C COMMON /BLOCKD/ ANGLEO,ANGLE1,ANGLE2,SANGLE
C
C COMMON /BLOCKG/ GAUSP(2)
C COMMON /INITL/ AO,B0,A1,B1
C COMMON /RESULT/ YLOAD,YF0,THRST,RESIS
C
C COMMON /TOLRN/ EPSKKG,EPSEQV,EPSWRK,EPSPRC,ZOMAX,NEPSJ
C COMMON /STRES/ P,TAUC,TAUP
C COMMON /INTF/ SUMINT,NINT,IEQV,KOUNT
C
C CHARACTER CASE'S
C INTEGER STRESS
C DIMENSION S(4)
C
C PL=3 141592653589793D0
C GAUSP(1)=0 577350269189626D0
C GAUSP(2)=GAUSP(1)
C PERCNT=0 01D0
C CASE='RIGID'
C
C OPEN ('15 FILE='test.in' STATUS='old')
C
C Read outer radius of the tire

```

```

C lug height,
C b in Bekker's equation,
C and tire tread.
C
C all units are in cm.
C
C READ (15,*) RLLG,HTLUG,BB,WIDTH
C RLLG=RLLG*PERCNT
C HTLUG=HTLUG*PERCNT
C BB=BB*PERCNT
C WIDTH=WIDTH*PERCNT
C
C Read slip (in percent),
C load applied (in kN),
C ratio of lug area to carcass area
C
C READ (15,*) SLIP,YLOAD,RATIO
C SLIP=SLIP*PERCNT
C
C Read Kc,
C Kphi,
C n,
C in Bekker's pressure-sinkage relationship
C K0,
C Au,
C in repetitive loading relationship
C
C READ (15,*) KC,KPHI,NN,K0,AU
C
C Read cohesion (in kPa),
C angle of shearing resistance (in degree)
C shear modulus (in cm),
C in exponential shear stress-shear displacement relationship
C
C and
C As,
C Bs,
C Ks,
C in slip sinkage-shear displacement relationship
C for rubber terrain shearing
C
C READ (15,*) CRUB,PHIRUB,PKRUB,ARUB,BRUB,PK2RUB
C PKRUB=PKRUB*PERCNT
C
C Read cohesion (in kPa),
C angle of shearing resistance (in degree),
C shear modulus (in cm),
C in exponential shear stress-shear displacement relationship
C
C and
C As

```

```

C      Bs.
C      Ks.
C      in slip sinkage-shear displacement relationship
C
C      for internal shearing
C
C      READ (15,*) CNT,PHINT,PKINT,AINT,BINT,PK2INT
C      PKINT=PKINT*PERCNT
C
C      Read allowable errors
C      for searching sinkage, EPSSKG,
C      for searching slip sinkage, EPSEQV,
C      for integrating equivalent work done, EPSWRK,
C      for integrating normal reactions, trust, and resistance, EPSFRC,
C      and integration step NEPSJ.
C
C      READ (15,*) EPSSKG,EPSEQV,EPSWRK,EPSFRC,NEPSJ
C
C      Read the lower and upper limits for sinkage,
C      A0, B0 in cm,
C      and the lower and upper limits for slip sinkage,
C      A1, B1 in cm.
C
C      READ (15,*) Z0,B0,A1,B1
C      A0=A0*PERCNT
C      B0=B0*PERCNT
C      A1=A1*PERCNT
C      B1=B1*PERCNT
C
C      Read control variable STRESS.
C      STRESS = 1: Stress distributions are wanted.
C      STRESS = 0: Stress distributions are not wanted.
C
C      READ (15,*) STRESS
C      CLOSE (15)
C
C      Do some initial calculations
C
C      K=KC/BB*KPHI
C      TANFIL=DTAN(PHIRUB*PI/180D0)
C      TANFIC=DTAN(PHINT*PI/180D0)
C      RCAR=RLUG-HTLUG
C
C      Search for sinkage and slip sinkage at wheel center.
C
C      CALL FNDSKG (CASE,SINK)
C
C      Write the outputs to the file TEST.OUT.
C
C      DBP=THRST-RESIS
C      OPEN (7,FILE='test.out')

```

```

WRITE (7,*) 'Normal reaction' YF0
WRITE (7,*) THRST = THRST
WRITE (7,*) RESIS = RESIS
WRITE (7,*) DBP = DBP
WRITE (7,*) SINKAGE = SINK
WRITE (7,*) SLIP_SINKAGE = Z0
CLOSE (7)
C
C      If stress distributions are wanted, then write them to STRESS OUT
C
C      IF (STRESS EQ 1) THEN
C        ZSINK=SINK
C        NP=100
C        CALL VALSET
C        & (RLUG,PKRUB,TANFIL,GAMMA0L,THET0L,THET1L,THET2L,DJEQVL,1)
C        OPEN (16,FILE='stress out')
C        P=0 D0
C        TAUP=0 D0
C        TAUC=0 D0
C        NINT=1
C        IEQV=2
C        SUMINT=0 D0
C        DGAM=(GAMMA0-GAMMA2)*0.01 D0
C        WRITE (16,10) GAMMA0,P,TAUP,TAUC
C        DO 1 I=1,NP
C          GAMMA=GAMMA2+DGAM*DBLE(NP-I)
C          CALL FSR (2,3,GAMMA, DGAM,S)
C          WRITE (16,10) GAMMA,P,TAUP,TAUC
C        1 CONTINUE
C        CLOSE (16)
C        10 FORMAT(4(G12.5))
C        ENDIF
C      STOP
C      END

```

```

C -----|||
C SUBROUTINE FNDSKG (CASE,X)
C
C This subroutine searches the sinkage at wheel center.
C
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C CHARACTER CASE*5
C
COMMON /BLOCKA/ PI,R,XL,Z0,WIDTH,THETA0,THETA1,THETA2,
& STHETA,CTHETA,GAMMA0,GAMMA1,GAMMA2,SGAMMA
C
COMMON /INITL/ AO,B0,A1,B1
COMMON /TOLRN/ EPSKKG,EPSEOV,EPSPWK,EPSPRC,ZOMAX,NEPSJ
C
C This method, developed by Songtao Chen, is a combination of secant,
C false position, bisection and quasi-Newton methods.
C
C Note: AO < B0, initial range of the solution
C K -----> no. of function calls.
C EPSKKG --> allowable sinkage error.
C MAXF -----> maximum no. of function call.
C X -----> solution.
C
FC=1D99
FD=1D99
MAXF=50
A=AO
B=B0
C
C Check the function values at given limits, and direct the
C search accordingly
C
KT=0
CALL INTFC (A,FA,KT,CASE)
CALL INTFC (B,FB,KT,CASE)
C
C If signs are different, then use secant--false position--bisection
C method, starting at 1
C
IF (FB*FA LT 0D0) GOTO 1
C
C If not, try to find the range that function changes its signs
C
2 CK=(FB+FA)/(B-A)
C=A+FA*CK
C=MIN(0.9D0*R,MAX(C,1D-6))
CALL INTFC (C,FC,KT,CASE)
C
C check convergence
C
IF (DABS(FC) LT EPSKKG) THEN
X=C

```

```

GOTO 5
ENDIF
IF (KT.GT.MAXF) THEN
WRITE (':') 'NO. OF ITERATIONS EXCEEDS MAX. NO. ALLOWED'
WRITE (':') 'IN SUBROUTINE FNDSKG'
STOP
ENDIF
C
TEMP1=FC*FB
TEMP2=FC*FA
IF ((TEMP1 LT 0D0) .OR. (TEMP2 LT 0D0)) THEN
IF (C LT A) THEN
B=A
FB=FA
A=C
FA=FC
ELSEIF (C.GT.A .AND. C LT B) THEN
IF (TEMP1 LT 0D0) THEN
A=C
FA=FC
ELSE
B=C
FB=FC
ENDIF
ELSEIF (C.GT.B) THEN
A=B
FA=FB
B=C
FB=FC
ENDIF
ENDIF
C
C If the range has been found, then use secant--false position--bisection
C method, starting at 1
C
GOTO 1
ENDIF
C
C No feasible range has been found, conduct quasi-Newton search
C
TEMPA=DABS(FA)
TEMPB=DABS(FB)
TEMPC=DABS(FC)
IF (TEMPA GT TEMF3) THEN
A=B
FA=FB
B=C
FB=FC
ELSE
B=A
FB=FA
A=C
FA=FC

```



```

COMMON /BLOCKX/ P1,R,XL,Z0,WIDTH,THETA0,THETA1,THETA2,
& STHETA,CTHETA,GAMA0,GAMA1,GAMA2,SGAMMA

DOUBLE PRECISION KC,KPHI,NN,KK,CC
COMMON /BLOCKY/ KC,KPHI,NN,KK,CC,CP,CP2,CP3,CP4,CP5,CP6,CP7,CP8,CP9,CP10,CP11,CP12,CP13,CP14,CP15,CP16,CP17,CP18,CP19,CP20,CP21,CP22,CP23,CP24,CP25,CP26,CP27,CP28,CP29,CP30,CP31,CP32,CP33,CP34,CP35,CP36,CP37,CP38,CP39,CP40,CP41,CP42,CP43,CP44,CP45,CP46,CP47,CP48,CP49,CP50,CP51,CP52,CP53,CP54,CP55,CP56,CP57,CP58,CP59,CP60,CP61,CP62,CP63,CP64,CP65,CP66,CP67,CP68,CP69,CP70,CP71,CP72,CP73,CP74,CP75,CP76,CP77,CP78,CP79,CP80,CP81,CP82,CP83,CP84,CP85,CP86,CP87,CP88,CP89,CP90,CP91,CP92,CP93,CP94,CP95,CP96,CP97,CP98,CP99,CP100,CP101,CP102,CP103,CP104,CP105,CP106,CP107,CP108,CP109,CP110,CP111,CP112,CP113,CP114,CP115,CP116,CP117,CP118,CP119,CP120,CP121,CP122,CP123,CP124,CP125,CP126,CP127,CP128,CP129,CP130,CP131,CP132,CP133,CP134,CP135,CP136,CP137,CP138,CP139,CP140,CP141,CP142,CP143,CP144,CP145,CP146,CP147,CP148,CP149,CP150,CP151,CP152,CP153,CP154,CP155,CP156,CP157,CP158,CP159,CP160,CP161,CP162,CP163,CP164,CP165,CP166,CP167,CP168,CP169,CP170,CP171,CP172,CP173,CP174,CP175,CP176,CP177,CP178,CP179,CP180,CP181,CP182,CP183,CP184,CP185,CP186,CP187,CP188,CP189,CP190,CP191,CP192,CP193,CP194,CP195,CP196,CP197,CP198,CP199,CP200,CP201,CP202,CP203,CP204,CP205,CP206,CP207,CP208,CP209,CP210,CP211,CP212,CP213,CP214,CP215,CP216,CP217,CP218,CP219,CP220,CP221,CP222,CP223,CP224,CP225,CP226,CP227,CP228,CP229,CP230,CP231,CP232,CP233,CP234,CP235,CP236,CP237,CP238,CP239,CP240,CP241,CP242,CP243,CP244,CP245,CP246,CP247,CP248,CP249,CP250,CP251,CP252,CP253,CP254,CP255,CP256,CP257,CP258,CP259,CP260,CP261,CP262,CP263,CP264,CP265,CP266,CP267,CP268,CP269,CP270,CP271,CP272,CP273,CP274,CP275,CP276,CP277,CP278,CP279,CP280,CP281,CP282,CP283,CP284,CP285,CP286,CP287,CP288,CP289,CP290,CP291,CP292,CP293,CP294,CP295,CP296,CP297,CP298,CP299,CP300,CP301,CP302,CP303,CP304,CP305,CP306,CP307,CP308,CP309,CP310,CP311,CP312,CP313,CP314,CP315,CP316,CP317,CP318,CP319,CP320,CP321,CP322,CP323,CP324,CP325,CP326,CP327,CP328,CP329,CP330,CP331,CP332,CP333,CP334,CP335,CP336,CP337,CP338,CP339,CP340,CP341,CP342,CP343,CP344,CP345,CP346,CP347,CP348,CP349,CP350,CP351,CP352,CP353,CP354,CP355,CP356,CP357,CP358,CP359,CP360,CP361,CP362,CP363,CP364,CP365,CP366,CP367,CP368,CP369,CP370,CP371,CP372,CP373,CP374,CP375,CP376,CP377,CP378,CP379,CP380,CP381,CP382,CP383,CP384,CP385,CP386,CP387,CP388,CP389,CP390,CP391,CP392,CP393,CP394,CP395,CP396,CP397,CP398,CP399,CP400,CP401,CP402,CP403,CP404,CP405,CP406,CP407,CP408,CP409,CP410,CP411,CP412,CP413,CP414,CP415,CP416,CP417,CP418,CP419,CP420,CP421,CP422,CP423,CP424,CP425,CP426,CP427,CP428,CP429,CP430,CP431,CP432,CP433,CP434,CP435,CP436,CP437,CP438,CP439,CP440,CP441,CP442,CP443,CP444,CP445,CP446,CP447,CP448,CP449,CP450,CP451,CP452,CP453,CP454,CP455,CP456,CP457,CP458,CP459,CP460,CP461,CP462,CP463,CP464,CP465,CP466,CP467,CP468,CP469,CP470,CP471,CP472,CP473,CP474,CP475,CP476,CP477,CP478,CP479,CP480,CP481,CP482,CP483,CP484,CP485,CP486,CP487,CP488,CP489,CP490,CP491,CP492,CP493,CP494,CP495,CP496,CP497,CP498,CP499,CP500,CP501,CP502,CP503,CP504,CP505,CP506,CP507,CP508,CP509,CP510,CP511,CP512,CP513,CP514,CP515,CP516,CP517,CP518,CP519,CP520,CP521,CP522,CP523,CP524,CP525,CP526,CP527,CP528,CP529,CP530,CP531,CP532,CP533,CP534,CP535,CP536,CP537,CP538,CP539,CP540,CP541,CP542,CP543,CP544,CP545,CP546,CP547,CP548,CP549,CP550,CP551,CP552,CP553,CP554,CP555,CP556,CP557,CP558,CP559,CP560,CP561,CP562,CP563,CP564,CP565,CP566,CP567,CP568,CP569,CP570,CP571,CP572,CP573,CP574,CP575,CP576,CP577,CP578,CP579,CP580,CP581,CP582,CP583,CP584,CP585,CP586,CP587,CP588,CP589,CP590,CP591,CP592,CP593,CP594,CP595,CP596,CP597,CP598,CP599,CP600,CP601,CP602,CP603,CP604,CP605,CP606,CP607,CP608,CP609,CP610,CP611,CP612,CP613,CP614,CP615,CP616,CP617,CP618,CP619,CP620,CP621,CP622,CP623,CP624,CP625,CP626,CP627,CP628,CP629,CP630,CP631,CP632,CP633,CP634,CP635,CP636,CP637,CP638,CP639,CP640,CP641,CP642,CP643,CP644,CP645,CP646,CP647,CP648,CP649,CP650,CP651,CP652,CP653,CP654,CP655,CP656,CP657,CP658,CP659,CP660,CP661,CP662,CP663,CP664,CP665,CP666,CP667,CP668,CP669,CP670,CP671,CP672,CP673,CP674,CP675,CP676,CP677,CP678,CP679,CP680,CP681,CP682,CP683,CP684,CP685,CP686,CP687,CP688,CP689,CP690,CP691,CP692,CP693,CP694,CP695,CP696,CP697,CP698,CP699,CP700,CP701,CP702,CP703,CP704,CP705,CP706,CP707,CP708,CP709,CP710,CP711,CP712,CP713,CP714,CP715,CP716,CP717,CP718,CP719,CP720,CP721,CP722,CP723,CP724,CP725,CP726,CP727,CP728,CP729,CP730,CP731,CP732,CP733,CP734,CP735,CP736,CP737,CP738,CP739,CP740,CP741,CP742,CP743,CP744,CP745,CP746,CP747,CP748,CP749,CP750,CP751,CP752,CP753,CP754,CP755,CP756,CP757,CP758,CP759,CP760,CP761,CP762,CP763,CP764,CP765,CP766,CP767,CP768,CP769,CP770,CP771,CP772,CP773,CP774,CP775,CP776,CP777,CP778,CP779,CP780,CP781,CP782,CP783,CP784,CP785,CP786,CP787,CP788,CP789,CP790,CP791,CP792,CP793,CP794,CP795,CP796,CP797,CP798,CP799,CP800,CP801,CP802,CP803,CP804,CP805,CP806,CP807,CP808,CP809,CP810,CP811,CP812,CP813,CP814,CP815,CP816,CP817,CP818,CP819,CP820,CP821,CP822,CP823,CP824,CP
```

```

C
C
C      set contact angles.
C
COMMON /BLOCKX/ PI,R,XL,Z0,WIDTH,THETA0,THETA1,THETA2,
&      STHETA,CTHETA,GAMMA0,GAMMA1,GAMMA2,SGAMMA
C
DOUBLE PRECISION KC,KPHI,NN,K0,KU
COMMON /BLOCKB/ KC,KPHI,NN,BB,K0,AU,ZU,KU,PU,SLIP,D,EOV,
&      ZSINK,TANPHI,PK,A7,B7,PK2
C
COMMON /BLOCKC/ RLUG,RCAR,HTLUG,GAMA0L,GAMA0C,GAMA1L,GAMA1C,
&      GAMA2L,GAMA2C,THETOL,THETOC,THET1L,THET1C,
&      THET2L,THET2C,RATIO,
&      CRUB,PHIRUB,PKRUB,ARL3,BRUB,PK2RUB,TANFIL,
&      CINT,PHIINT,PKINT,AINT,BINT,PK2INT,TANFIC,
&      ZOMAXL,ZOMAXC,DJEQVL,DJEQVC
C
COMMON /BLOCKD/ ANGLE0,ANGLE1,ANGLE2,SANGLE
COMMON /INTF / SUMINT,NINT,IEQV,KOUNT
C
C      set radius, contact angles, etc
C
R=V1
PK=V2
TANPHI=V3
GAMMA0=V4
THETA0=V5
SGAMMA=DSIN(11.314*GAMMA0)
STHETA=DSIN(11.314*THETA0)
CTHETA=DCOS(THETA0)
THETA1=V6
THETA2=V7
TEMP=DCOS(THETA1)
ZU=R*(TEMP*CTHETA)
C
C      set terrain constant.
C
IF (11.EQ.1) THEN
  K=KC/BB*KPHI
ELSE
  K=KC/WIDTH*KPHI
ENDIF
C
C      calculate some parameters for recovery.
C
IF (ZU.GT.0.D0) THEN
  PU=K*ZU**NN
  KU=K0+AU*ZU
  ZE=PUKU
ELSE
  PU=0.D0
  ZE=0.D0
ENDIF
C
C
C      set contact angles.
C
ANGLE0=THETA0
ANGLE1=THETA1
ANGLE2=THETA2
SANGLE=STHETA
NINT=11
C
IF (11.EQ.1) THEN
  GAMMA1=GAMA1L
  GAMMA2=GAMA2L
ELSE
  GAMMA1=GAMA1C
  GAMMA2=GAMA2C
ENDIF
C
C      calculate the equivalent shear displacement where normal pressure
C      starts declining.
C
PMAK=K*(R*(1.D0-CTHETA))**NN
DJ=R*(THETA0-(1.D0-SLIP)*STHETA)
CALL INTJ(D0,THETA0,50,N1,VINT,'RIGID')
TAUP=TANPHI*DEXP(DJ/PK)*VINT*P/PK
DJEQV=PK*DLOG(1.D0-TAUP/(PMAK*TANPHI))
C
RETURN
END
C
SUBROUTINE FNDEQV
C
C      This subroutine searches for the slip-sinkage at wheel center.
C
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C      CHARACTER CASE*5
C
COMMON /INITL/ A0,B0,A1,B1
COMMON /TOLRN/ EPSKG,EPSEQV,EPWRK,EPFRC,ZOMAX,NEPSJ
C
C      This method, developed by Songtao Chen is a combination of secant,
C      false position, bisection and quasi-Newton methods.
C
C      Note: A1 < B1, initial range of the solution.
C      K -----> no. of function calls.
C      EPSEQV ---> allowable equivalent work error.
C      MAXF -----> maximum no. of function call.
C      X -----> solution.
C
FC=1D99
FD=1D99
MAXF=50

```

```

A=A1
B=B1
CASE='RIGID'
C
C Check the function values at given limits, and direct the
C search as appropriate.
C
KT=0
CALL EQUIV (A,FA,KT,CASE)
CALL EQUIV (B,FB,KT,CASE)
C
C If signs are different, then use secant--false position--bisection
C method, starting at 1.
C
IF (FB*FA.LT.0.00) GOTO 1
C
C If not, try to find the range that function changes its signs.
C
2 CK=(FB-FA)/(B-A)
C=A-FA*CK
C=MIN(ZOMAX,MAX(C,1D-9))
CALL EQUIV (C,FC,KT,CASE)
C
C check convergence.
C
IF (DABS(FC).LT.EPSEQV) THEN
X=C
GOTO 5
ENDIF
C
IF (KT.GT.MAXF) THEN
WRITE ('.') 'NO OF ITERATIONS EXCEEDS MAX NO ALLOWED'
WRITE ('.') 'IN SUBROUTINE FNDEQV'
STOP
ENDIF
C
TEMP1=FC*FB
TEMP2=FC*FA
IF ((TEMP1.LT.0.00) OR (TEMP2.LT.0.00)) THEN
IF (C.LT.A) THEN
B=A
FB=FA
A=C
FA=FC
ELSEIF (C.GT.A.AND.C.LT.B) THEN
F (TEMP1.LT.0.00) THEN
A=C
FA=FC
ELSE
B=C
FB=FC
ENDIF
ELSEIF (C.GT.B) THEN

```

```

A=B
FA=FB
B=C
FB=FC
ENDIF
C
C If the range has been found, then use secant--false position--bisection
C method, starting at 1.
C
GOTO 1
ENDIF
C
C No leasible range has been found, conduct quasi-Newton search.
C
TEMPA=DABS(FA)
TEMPB=DABS(FB)
TEMPC=DABS(FC)
IF (TEMPA.GT.TEMPB) THEN
A=B
FA=FB
B=C
FB=FC
ELSE
B=A
FB=FA
A=C
FA=FC
ENDIF
GOTO 2
C
1 CK=(FB-FA)/(B-A)
C=A-FA*CK
C=MIN(ZOMAX,MAX(C,1D-9))
CALL EQUIV (C,FC,KT,CASE)
C
C check convergence.
C
IF (DABS(FC).LT.EPSEQV) THEN
X=C
GOTO 5
ENDIF
C
C use secant--false position method
C
IF (FC*FA.GT.0.D0) THEN
CK=(FC-FA)/(C-A)
D=A-FA*CK
IF ((D.GT.C).AND.(D.LT.B)) THEN
D=MIN(ZOMAX,MAX(D,1D-9))
CALL EQUIV (D,FD,KT,CASE)
IF (FD*FB.GT.0.D0) THEN
A=C

```



```

      B=D
      FB=FC
      ELSE
      A=C
      FA=FC
      ENDIF
C
      IF (KT.GT.MAXF) THEN
      WRITE ('.') 'NO. OF ITERATIONS EXCEEDS MAX. NO. ALLOWED'
      WRITE ('.') 'IN SUBROUTINE FNDEQV'
      STOP
      ENDIF
      GOTO 1
C
      5 Z0=X
      RETURN
      END
C
      .....
      SUBROUTINE EQUIV (Z00,YF,KT,CASE)
C
C      This subroutine calculates the equivalent work done.
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*5 CASE
C
      COMMON /BLOCKA/ P1,R,XL,Z0,WIDTH,THETA0,THETA1,THETA2,
&
&      STHETA,CTHETA,GAMMA0,GAMMA1,GAMMA2,SGAMMA
C
      DOUBLE PRECISION KC,KPHI,NN,K,K0,KU
      COMMON /BLOCKB/ KC,KPHI,NN,BB,K,K0,AU,ZU,KU,PU,SLIP,D,JEQV,
&
&      ZSINK,TANPHI,PK,A7,B7,PK2
C
      COMMON /BLOCKC/ RLUG,RCAR,HTLUG,GAMA0L,GAMA0C,GAMA1L,GAMA1C,
&
&      GAMA2L,GAMA2C,THET0L,THET0C,THET1L,THET1C,
&
&      THET2L,THET2C,RATIO,
&
&      CRUB,PHIRUB,PKRUB,ARUB,BRUB,PK2RUB,TANFIL,
&
&      CINT,PHINT,PKINT,AINT,BINT,PK2INT,TANFIC,
&
&      ZOMAXL,ZOMAXC,DJEQVL,DJEQVC
C
      COMMON /BLOCKD/ ANGLE0,ANGLE1,ANGLE2,SANGLE
C
      COMMON /INTF / SUMINT,NINT,IEQV,KOUNT
      COMMON /TOLRN/ EPSKG,EPSEQV,EPSWRK,EPSFRC,ZOMAX,NEPSJ
C
      DIMENSION KFLAG(4)
      COMMON /INTGRT/ M,KFLAG
C
      DIMENSION S(4),S0(4)
C
      KT=KT+1
      WRITE ('.')
      iteration #',KT

```

```

      B=D
      FA=FC
      FB=FD
      ELSE
      A=D
      FA=FD
      ENDIF
      ELSE
      A=C
      FA=FC
      ENDIF
      ELSE
      CK=(FB-FC)/(B-C)
      D=C-FC/CK
      IF ((D.GT.A) .AND. (D.LT.C)) THEN
      D=MIN(ZOMAX,MAX(D,1D-9))
      CALL EQUIV (D,FD,KT,CASE)
      IF (FD*FA.GT.0.D0) THEN
      A=D
      B=C
      FA=FD
      FB=FC
      ELSE
      B=D
      FB=FD
      ENDIF
      ELSE
      B=C
      FB=FC
      ENDIF
      ENDIF
C
C      check convergence.
C
C      IF (DABS(FD)/LT.EPSEQV) THEN
      X=D
      GOTO 5
      ENDIF
C
C      use bisection method.
C
C      C=0.5D0*(A+B)
      C=MIN(ZOMAX,MAX(C,1D-9))
      CALL EQUIV (C,FC,KT,CASE)
      IF (DABS(FC)/LT.EPSEQV) THEN
      X=C
      GOTO 5
      ENDIF
C
C      reduce search range.
C
C      IF (FC*FALT.0D0) THEN

```

```

Z0=Z00
C
C get the difference in equivalent work done between along
C predicted slip-sinkage and geometrically allowable slip-sinkage
C on lugs.
C
CALL SETVAL (RLUG,THETOL,TANFIL,PKRUB,ARUB,BRUB,PK2RUB,DJEQVL,1)
CALL INTEQV (2,THETOL,THET2L,S,CASE)
C
DO 1 I=1,2
  SQ(I)=S(I)
1 CONTINUE
C
C get the difference in equivalent work done between along
C predicted slip-sinkage and geometrically allowable slip-sinkage
C on carcass.
C
CALL SETVAL (RCAR,THETOC,THET2C,TANFIC,PKINT,AINT,BINT,PK2INT,DJEQVC,2)
CALL INTEQV (2,THETOC,THET2C,S,CASE)
C
C compute the total difference in equivalent work done.
C
YF=RATIO*(S(1)-S(2)) + (1-D0-RATIO)*(S(1)-S(2))
WRITE (*,*) YF
RETURN
END
C
SUBROUTINE SETVAL (V1,V2,V3,V4,V5,V6,V7,V8,I1)
C
C This subroutine sets the proper parameters according to the
C control variable I1
C
I1 = 1 is for rubber-terrain shearing
I1 = 2 is for internal shearing
C
C for integrating the equivalent work done
C
CALL SETVAL(RLUG,THETOL,TANFIL,PKRUB,ARUB,BRUB,PK2RUB,DJEQVL,1)
CALL SETVAL(RCAR,THETOC,THET2C,TANFIC,PKINT,AINT,BINT,PK2INT,DJEQVC,2)
C
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
COMMON /BLOCKX/ PI,R,XL,Z0,WIDTH,THETA0,THETA1,THETA2
&
  STHETA,CTHETA,GAMMA0,GAMMA1,GAMMA2 SGAMMA
C
DOUBLE PRECISION KC,KPHI,NN,K,K0,KU
COMMON /BLOCKB/ KC,KPHI,NN,BB,K,K0,AU,ZU,KU,PU,SLIP,DJEQV,
&
  ZSNK,TANPHI,PK,A7,B7,PK2
C
COMMON /BLOCKC/ RLUG,RCAR,HTLUG,GAMA0L,GAMA0C,GAMA1L,GAMA1C,
&
  GAMA2L,GAMA2C,THETOL,THETOC,THET1L,THET1C,
&
  THET2L,THET2C,RATIO,
&
  CRUB,PHIRUB,PKRUB,ARUB,BRUB,PK2RUB,TANFIL,
&
  CINT,PHINT,PKINT,AINT,BINT,PK2INT,TANFIC,
&
  ZOMAXL,ZOMAXC,DJEQVL,DJEQVC
C
COMMON /INTF / SUMINT,NINT,IEQV,KOUNT
C
COMMON /BLOCKD/ ANGLE0,ANGLE1,ANGLE2,SANGLE
C
C set radius and contact angles.
C
R=V1
THETA0=V2
CTHETA=DCOS(THETA0)
STHETA=DSIN(THETA0)
IF (I1.EQ.1) THEN
  GAMA0L=DACOS((R*CTHETA-Z0)/R)
  GAMMA0=GAMA0L
ELSE
  GAMA0C=THETA0+DASIN((Z0*STHETA+XL*CTHETA)/R)
C
  Using ZOMAXC as ZS0
C
  GAMMA0=GAMA0C
  ZOMAXC=(R*DCOS(GAMMA0)+Z0)/CTHETA - R
  ENDIF
C
  SGAMMA=DSIN(GAMMA0)
  IF (I1.EQ.1) XL=R*(SGAMMA*STHETA)
  GAMMA1=DASIN(XL/R)
  THETA1=DATAN(XL/(R+Z0))
  TEMP=DCOS(THETA1)
C
  set terrain constant
C
  IF (I1.EQ.1) THEN
    K=KC/BB+KPHI
  ELSE
    K=KC/WIDTH+KPHI
  ENDIF
C
  calculate some parameters for recovery
C
  ZU=R*(TEMP*CTHETA)
  IF (ZU>GT.0.D0) THEN
    PU=K*ZU**NN
    KU=K0+AU*ZU
    ZE=PUKU
  ELSE
    PU=0.D0
    ZE=0.D0
  ENDIF
C

```

```

C      set contact angles
C
      TANPHI=V3
      PK=V4
      THETA2=DACOS(TEMP-ZE/R)
      GAMMA2=THETA2+DASIN((20'DSIN(THETA2)+XL'DCOS(THETA2))/R)

C
      IF (I1.EQ.1) THEN
        THET1L=THETA1
        THET2L=THETA2
        GAMMA1L=GAMMA1
        GAMMA2L=GAMMA2
      ELSE
        THET1C=THETA1
        THET2C=THETA2
        GAMMA1C=GAMMA1
        GAMMA2C=GAMMA2
      ENDIF

C      set slip-sinkage parameters
C
      A7=V5
      B7=V6
      PK2=V7
      NINT=I1

C      set contact angles and calculate the equivalent shear displacement
C      where normal pressure starts declining.
C
      ANGLE0=THETA0
      ANGLE1=THETA1
      ANGLE2=THETA2
      SANGLE=STHETA
      PMAX=K'R*(1.D0-GTHETA)**NN
      DJ=R*(THETA0-(1D0-SLIP)*STHETA)
      CALL INTJ(DD0,THETA0,50,N1,VINT,'RIGID')
      ZSP=B7'DEXP(-DJ/PK2)*VINT*R/PK2
      DJEQV=-PK2'DLOG(1.D0-ZSP/(PMAX*B7))

C
      RETURN
      END
C

```

```

C-----
SUBROUTINE SETSIM(N,A,B,S,CASE)
C
C This is a Simpson integration procedure.
C It integrates the stress distributions to obtain
C total vertical reaction, thrust and motion resistance.
C
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C CHARACTER'S CASE
C DIMENSION KFLAG(4),S(4),T(4),Q(4),SO(4)
C COMMON /INTGRT/ M,KFLAG
C COMMON /INTF / SUMINT,NINT,IEOV,KOUNT
C COMMON /TOLRV EPSSKG,EPSEQV,EPSWRK,EPSFRC,ZOMAX,NEPSJ
C
C M=1
C DO 10 I=1,N
C   SQ(I)=OD0
C   KFLAG(I)=0
C 10 CONTINUE
C
C evaluate the function at given limits according to
C the bre operating modes.
C
C IF (CASE.EQ.'RIGID') THEN
C   STEP=0D0
C   CALL FSR(-1,N,A,STEP,S)
C   STEP=B-A
C   CALL FSR(-2,N,B,STEP,Q)
C ELSE
C   STEP=0D0
C   CALL FSE(-1,N,A,STEP,S)
C   STEP=B-A
C   CALL FSE(-2,N,B,STEP,Q)
C ENDIF
C
C set initial step and reference
C
C H=0.5D0*(B-A)
C C=DABS(A)+DABS(B)
C
C DO 4 J=1,N
C   T(J)=H*(S(I)+Q(I))
C 4 CONTINUE
C
C 1 X=A+H
C DO 5 L=1,N
C   Q(L)=0.5D0*T(L)
C 5 CONTINUE
C
C evaluate the functions at given steps
C
C DO 2 L=1,M

```



```
SJ-SJ.P
2  CONTINUE
  S-S.SJ
1  CONTINUE
  S-S'DIVLF
C
  RETURN
  END
C
```

```

C-----
C SUBROUTINE RWRK (M,N,THETA,S,STEP)
C
C This subroutine computes the equivalent work done for rigid wheel.
C
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C DIMENSION S(N)
C
C COMMON /BLOCKA/ P1,R,XL,Z0,WIDTH,T1,ETA0,THETA1,THETA2,
C & STHETA,CTHETA,GAMMA0,GAMMA1,GAMMA2,SGAMMA
C
C DOUBLE PRECISION KC,KPHI,NN,K,K0,KU
C COMMON /BLOCKB/ KC,KPHI,NN,BB,K,K0,AU,ZU,KU,PU,SLIP,D,JEQV,
C & ZSINK,TANPHI,PK,A7,B7,PK2
C
C COMMON /BLOCKC/ RLUG,RCAR,HTLUG,GAMA0L,GAMA0C,GAMA1L,GAMA1C,
C & GAMA2L,GAMA2C,THETOL,THETOC,THET1L,THET1C,
C & THET2L,THET2C,RATIO,
C & CRUB,PHIRUB,PKRUB,APUB,BRUB,PK2RUB,TANFIL,
C & CINT,PHINT,PKINT,AINT,BINT,PK2INT,TANFIC,
C & ZOMAXL,ZOMAXC,DJEQVL,DJEQVC
C
C COMMON /INTF / SUMINT,NINT,IEQV,KOUNT
C COMMON /TOLRV / EPSKGE,PEQOV,EPWRK,EPFRK,ZOMAX,NEPSJ
C
C TEMPC=DCOS(THETA)
C TEMPS=DSIN(THETA)
C TEMPO=THETA0-THETA
C GAMMA=THETA + DASIN((Z0*TEMPS+XL*TEMPC)/R)
C
C calculate sinkage.
C
C Z=R*(TEMPC-CTHETA)
C
C Rear recovery & virgin loading, calculate the normal pressure
C
C IF ((THETA.LT.THETA1) .AND. (THETA.GE.THETA2)) THEN
C   P=PU-KU*(ZU-Z)
C   P=MAX(P,0.D0)
C ELSE
C   IF (Z.GT.0.D0) THEN
C     P=K*Z**NN
C   ELSE
C     P=0.D0
C   ENDIF
C ENDIF
C
C Calculate the slip-sinkage and work done.
C
C DJ=R*(TEMPO*(1.D0-SLIP)*(STHETA-TEMPS))
C ZS=(A7+B7*P)**(1.D0-DEXPI-DJ/PK2))
C ZSP=(R*DCOS(GAMMA)+Z0)/TEMPC - R
C
C IF (NINT.EQ.2) THEN
C   ZSP=ZSP-ZOMAXC
C ENDIF
C S(1)=P*ZS
C S(2)=P*ZSP
C
C S(1): work done along predicted slip-sinkage, ZS.
C S(2): work done along geometrically allowable slip-sinkage, ZSP.
C
C RETURN
C END
C-----
C SUBROUTINE EWRK(M,N,THETA,S,STEP)
C
C This subroutine computes
C the equivalent work done for elastic tire.
C
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C WRITE(,' ') 'Subroutine EWRK is not implemented'
C RETURN
C END
C-----
C SUBROUTINE EQVLR (X1,PJ)
C
C This subroutine calculates the equivalent shear displacement
C for rigid wheel.
C
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C COMMON /BLOCKA/ P1,R,XL,Z0,WIDTH,T1,ETA0,THETA1,THETA2,
C & STHETA,CTHETA,GAMMA0,GAMMA1,GAMMA2,SGAMMA
C
C DOUBLE PRECISION KC,KPHI,NN,K,K0,KU
C COMMON /BLOCKB/ KC,KPHI,NN,BB,K,K0,AU,ZU,KU,PU,SLIP,D,JEQV,
C & ZSINK,TANPHI,PK,A7,B7,PK2
C
C COMMON /INTF / SUMINT,NINT,IEQV,KOUNT
C
C COMMON /BLOCKD/ ANGLE0,ANGLE1,ANGLE2,SANGLE
C
C TEMPC=DCOS(X1)
C TEMPS=DSIN(X1)
C
C Calculate sinkage and then
C normal pressure using Bekker's equation.
C
C Z=R*(TEMPC-CTHETA)
C
C Rear recovery & virgin loading

```

```

      & CRUB,PHIRUB,PKRUB,ARUB,BRUB,PK2RUB,TANFIL,
      & CINT,PHINT,PKINT,AINI,BINT,PK2INT,TANFIC,
      & ZOMAXL,ZOMAXC,DJEQVL,DJEQVC
C
      COMMON /STRES/ P,TAUC,TAUP
      COMMON /INTF / SUMINT,NINT,IEQV,KOUNT
C
      TEMPC=DCOS(GAMMA)
      TEMPS=DSIN(GAMMA)
C
      Calculate contact angle and sinkage.
C
      THETA=ATAN((R*TEMPS-XL)/(R*TEMPC+Z0))
      Z=R*(DCOS(THETA)*CTHETA)
C
      Rear recovery & virgin loading
C
      IF ((GAMMA LT 0D0) .AND. (GAMMA GE GAMMA2)) THEN
        P=PU-KU*(ZU-Z)
        P=MAX(P,0D0)
      ELSE
        IF (Z GT 0D0) THEN
          P=K*Z**NN
        ELSE
          P=0D0
        ENDIF
      ENDIF
C
      calculate shear displacement and cohesive shear stress
C
      DJ=R*(THETA-THETA*(1D0-SLIP)*(STHETA-DSIN(THETA)))
      IF (NINT EQ 1) THEN
        TAUC=CRUB*(1D0-DEXP(-DJ*PKRUB))*CINT*(1D0-DEXP(-DJ*PKINT))
      ELSE
        TAUC=0D0
      ENDIF
C
      calculate shear stress using exponential equation
C
      IF (ABS(M) GT 1) THEN
        GAMMAJ=GAMMA-STEP
        THETAJ=ATAN((R*DSIN(GAMMAJ)-XL)/(R*DCOS(GAMMAJ)+Z0))
      ELSE
        SUMINT=0D0
        IF (M EQ 1) THETAJ=THETA
      ENDIF
C
      IF (GAMMA GE GAMMA1) THEN
        IF (M NE -1) THEN
          KOUNT=KOUNT+1
          IF (KOUNT LT NEPSJ) THEN
            NDIV=.2**(NEPSJ-KOUNT)
          ENDIF
        ENDIF
      ENDIF

```

```

      IF (X1 LT ANGLE1) .AND. (X1 GE ANGLE2)) THEN
        P=PU-KU*(ZU-Z)
        P=MAX(P,0D0)
      ELSE
        IF (Z GT 0D0) THEN
          P=K*Z**NN
        ELSE
          P=0D0
        ENDIF
      ENDIF
C
      calculate shear displacement and equivalent shear displacement.
C
      DJ=R*(ANGLE0-X1*(1D0-SLIP)*(SANGLE-TEMPS))
      IF (IEQV EQ 1) THEN
        PJ=P*DEXP(DJ*PK2)*(1D0-(1D0-SLIP)*TEMPC)
      ELSE
        PJ=P*DEXP(DJ*PK1)*(1D0-(1D0-SLIP)*TEMPC)
      ENDIF
C
      RETURN
      END
C
      SUBROUTINE EQVUE(XI,PJ)
C
      This subroutine calculates the equivalent shear displacement
      for elastic tire
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      WRITE (*,*) 'Subroutine EQVUE is not implemented'
      RETURN
      END
C
      SUBROUTINE FSR (MN,GAMMA,STEP,S)
C
      This subroutine computes the normal pressure and shear stress
      in the contact area for rigid wheel
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION S(N)
C
      COMMON BLOCKA PIR,XI,Z0,WIDTH,THETA0,THETA1,THETA2
      & STHETA,CTHETA,GAMMA0,GAMMA1,GAMMA2,SGAMMA
C
      DOUBLE PRECISION KC,KPHI,NN,K,K0,KU
      COMMON BLOCKB KC,KPHI,NN,BB,K,K0,AU,ZU,KU,PU,SLIP,DJEQV
      & ZSINK,TANPHI,PK,A7,B7,PK2
C
      COMMON BLOCKC RLUG,RCAR,HTLUG,GAMA0L,GAMA0C,GAMA1L,GAMA1C
      & GAMA2L,GAMA2C,THETOL,THETOC,THET1L,THET1C,
      & THET2L,THET2C,RATIO

```



```

ELSE
  NDIV=1
ENDIF
CALL INTX(THETA, THETAJ, NDIV, N1, VINT, RIGID)
ELSE
  VINT=DD0
ENDIF

C
SUMINT=SUMINT+VINT*R
TAUP=TANPHI*DEXP(-DJ/PK)*SUMINT/PK
ELSE
  DJ=DJEQV+R*(-THETA*(1D0-SLIP)*(-DSIN(THETA)))
  TAUP=TANPHI*P*(1D0-DEXP(-DJ/PK))
ENDIF

C
C S(1) Normal reaction from COS component of pressure
C + Normal reaction from SIN component of shear stress
C S(2) Thrust from COS component of shear stress
C S(3) Resistant from SIN component of pressure
C
S(1)=P*TEMPC+(TAUP+TAUC)*TEMPS
S(2)=(TAUP+TAUC)*TEMPC
S(3)=P*TEMPS

C
RETURN
END

C .....
SUBROUTINE FSE(M,N,GAMMA,STEP,S)
C
C This subroutine computes the normal pressure and shear stress
C in the contact area for elastic tire
C
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C WRITE ('.') 'Subroutine FSE is not implemented'
RETURN
END

```

Appendix D

Computer Source Codes for Simulation of Tire-Soft Terrain Interaction — Finite Element Model

Program TTIM is the finite element tire-terrain interaction model presented in this thesis.

Program INTERP is designed to interpolate the output files from TTIM to obtain more accurate tractive performance and tire deformations.

Program FNLSTR is designed to obtain the final normal pressure and shear stress distributions.

```

C .....
C      Tire-Terrain Interaction Model (TTIM)
C      by
C      Songliao Chen
C      Transport Technology Research Laboratory
C      Carleton University
C .....
C      (Ver. 1.0, December, 1991)
C .....
C      Input data file (1) --> gene3d.in
C      Input data file (2) --> terrain.dat
C      Output data file (n) --> ttim???.out
C .....
C
C      To understand this software, knowledge of three-dimensional
C      linear elastic finite element method is essential.
C
C      Understanding of the bevanmeter technique and its formulation are
C      also very important.
C .....
C      I M P L I C I T   R E A L * 8   ( A - H , O - Z )
C .....
C      PARAMETER(NP/MA=1300,NEMAX=600,NNG/DE=8,NDDOF=3,NMMAX=3)
C      PARAMETER(IKV=60000,ILOAD=NP/MA*NDDOF)
C      COMMON /ELNOD/  LNODE(NEMAX,NNODE)
C      COMMON /COORD3/ X(NP/MA),Y(NP/MA),Z(NP/MA)
C      COMMON /COORD3/ XU(NP/MA),YU(NP/MA),ZU(NP/MA)
C      COMMON /BOUND/  NBC(NP/MA,NDDOF)
C      COMMON /MATER/  YOUNG(NM/MA),POISS(NM/MA),NMAT(NEMAX)
C      COMMON /MEINF/  NP,NE,NM,NB
C      REAL*8      KC,KPHI,NN
C      COMMON /TERAIN/ KC,KPHI,NN,BB,A0,AU
C      COMMON /COEFFS/ COEFF(10,3),SAMP(2,2),NYCF(3),NGP
C      REAL*8      KINT,KRUB
C      COMMON /SHRINF/ CINT,TANFIL,KINT,CRUB,TANFIR,KRUB
C      COMMON /SURFAC/ YSURF,YMIN,XBOT,RBAR,THETA0,THETA1,DTHETA
C      COMMON /OTHER/  PI,RATIO,SLIP,ANG,PI2,NPSC
C      COMMON /SHRTP/  XBOTO,YBOTO,NSHTP,NSLPSK
C      COMMON /PRFRM/  TOTFY,THRST,RESIS,ZSNK,KSRC
C      COMMON /SKGPN/  A1AR,B1BR,P1PKR,EPSSKG
C      REAL*8      LAMDA
C      COMMON /SRCH/  ERR(3),EPS,LAMDA,GAMMA,CRTREF,
C      &      NEPS,IEND,ITER,ITER
C .....
C      REAL*8 BEE(6,24),BTD(24,24),COORD(8,3),DEE(6,6)
C      REAL*8 DER(3,8),DERIV(3,8),DBEE(6,24),EKIM(24,24)
C      REAL*8 TKV(IKV),EKMSAV(NEMAX,300)
C      REAL*8 DISPT(ILOAD),OLOAD(ILOAD)
C      REAL*8 FORCE(ILOAD),FORCE1(ILOAD)
C      REAL*8 XJAC(3,3),XJAC1(3,3)
C .....
C      INTEGER IG(NEMAX,24),KDIAG(ILOAD)
C      CHARACTER FLNM*12
C      DATA /BTD8,KIM,BT,DOF/4*24/,  IDEE,IBEE,DBEE,IBV4*8/
C      DATA /COORD,NOD/2*8/,  LAC,LAC1,IDER,IDERV,ITS*3/
C      DATA /IDER2,NGP/2*2/
C .....
C      Set Constants
C .....
C      IEND=0
C      CALL GAUSS
C .....
C      Read Input Data
C .....
C      WRITE (6,*) '*** T T I M version 1.0 ***'
C      WRITE (6,*) ' -> Reading input data'
C      CALL DATAIN(YLOAD,DINCR,CF1,CF2)
C .....
C      CALL SETBC
C .....
C      Calculate Maximum Diagonal
C      and Check if it will be overflow
C .....
C      DO 10 I=1,NB
C      KDIAG(I)=0
C      10 CONTINUE
C      DO 15 IE=1,NE
C      CALL IGSET(IE,IG)
C      CALL FKDIAG(KDIAG,IG,DOF,IE,NEMAX)
C      15 CONTINUE
C      KDIAG(1)=1
C      DO 20 I=2,NB
C      KDIAG(I)=KDIAG(I)+KDIAG(I-1)
C      20 CONTINUE
C      NKT=KDIAG(NB)
C      IF (NKT.GT.IKV) THEN
C      WRITE(6,900) NKT,IKV
C      900 FORMAT('TTIM ERROR. KDIAG(NB)=',I6,' > IKV=',I6)
C      STOP
C      ENDIF
C .....
C      Calculate the increments
C .....
C      CALL NULVEC('LOAD,NB)
C .....
C      CALL FNDBOT(NP,X,Y,Z,XBOT,YMIN,ZBOT)
C .....
C      INCR=0
C      80 INCR=INCR+1
C      YSURF=YSURF+DINCR
C      WRITE (6,*) ' ... INCREMENT #',INCR
C      CALL NULVEC(DISPT,NB)

```

```

C-----
C      CALL NULVEC(FORCE,NB)
C      Element Stiffness Integration and Save in EKMSAV
C-----
C      WRITE (6,*) '-> Calculating element stiffness'
C      E=1,DO
C      V=POISS(1)
C      CALL FORMQ3(DEE,IDE,E,V)
C      DO 40 IE=1,NE
C      CALL ECOORD(IE,COORD,COORD,X,Y,Z)
C      CALL NULL(EKM,IKM,IDO,IDO)
C      -- Gauss Integration Loop
C      DO 45 I=1,NGP
C      DO 45 J=1,NGP
C      DO 45 K=1,NGP
C      -- Stiffness Matrix
C      CALL FMLIN3(IDER,IDER,I,J,K)
C      CALL MATMUL(IDER,COORD,XJAC,IT,NOD,IT)
C      CALL TREEX3(XJAC,UAC,XJAC1,UAC1,DET,IE)
C      CALL MATMUL(XJAC1,DER,DERIV,IT,IT,NOD)
C      CALL NULL(BEE,IBEE,IH,IDO)
C      CALL FORMB3(BEE,IBEE,DERIV,IDERIV,NOD)
C      CALL MATMUL(DEE,BEE,DBEE,IH,IH,IDO)
C      QUOT=DET*SAMP1(1,2)*SAMP1(2,2)*SAMP1(K,2)
C      CALL BTDBCC(BTDB,BEE,DBEE,IBTDB,IBEE,IDBEE,QUOT)
C      CALL MATADD(EKM,IKM,BTDB,IBTDB,IDO,IDO)
C      45 CONTINUE
C      CALL SAVEKME(EKM,IKM,EKMSAV,NEMAX)
C      40 CONTINUE
C-----
C      Update elastic properties, coordinates and forces
C-----
C      199 ITER=0
C      200 CONTINUE
C      WRITE (6,*) '-> Updating'
C      CALL UPDATE(INCR,DISPT,FORCE,FORCE1,DINCR)
C-----
C      Assemble global stiffness matrix
C-----
C      WRITE (6,*) '-> Assembling global stiffness matrix'
C      CALL NULVEC(TKV,NKT)
C      DO 50 IE=1,NE
C      CALL RESERME(EKM,IKM,EKMSAV,NEMAX)
C      E=YOUNG(NMAT(IE))
C      CALL MSMULT(EKM,IKM,E,IDO,IDO)
C      CALL FSPARV(TKV,EKM,IKM,IG,KDIAG,IDO,E,NEMAX)
C      50 CONTINUE
C-----
C      Solve Linear Equations by Skyline Method
C-----
C      WRITE (6,*) '-> Solving linear equations'
C      CALL SPARIN(TKV,NB,KDIAG)
C-----
C      -- subtract the old load, OLOAD,
C      -- from the current load, FORCE,
C      -- to get incremental load.
C      CALL SUBVEC(FORCE,OLOAD,FORCE,NB)
C      -- loop here ensures that the tire
C      -- is in contact with the terrain
C      -- take the difference between the
C      -- current load, FORCE, and the load
C      -- from last iteration, FORCE1, and
C      -- save it in DISPT.
C      CALL SUBVEC(FORCE,FORCE1,DISPT,NB)
C      -- scale DISPT using parameter, lamda,
C      -- where 0.001 < lamda < 0.999
C      CALL MULVEC(DISPT,DISPT,LAMDA,NB)
C      -- subtract the above amount, DISPT,
C      -- from the current load, FORCE, and
C      -- save it in FORCE.
C      CALL SUBVEC(FORCE,DISPT,FORCE,NB)
C      -- assign the FORCE to DISPT for
C      -- solving equations.
C      CALL ASNVEC(DISPT,FORCE,NB)
C      the above procedure can be expressed as:
C      DISPT(I)=FORCE(I) - (FORCE(I)-FORCE1(I))*LAMDA
C      CALL SPABAC(TKV,DISPT,NB,KDIAG)
C      -- check whether the tire is in
C      -- contact with terrain or not
C      CALL CHKCONT(Y,DISPT,NP,ICNT)
C      -- if not in contact, larger
C      -- scale factor is used to reduce
C      -- the applied load, FORCE.
C      IF (ICNT EQ.0) THEN
C      LAMDA=DSORT(LAMDA)
C      IF (LAMDA.GT.0.999999DO) THEN
C      WRITE (6,*) 'LAMDA IS',LAMDA
C      STOP
C      ENDIF
C      WRITE (6,*) '... not in contact !!!'
C      GOTO 75
C      ENDF
C-----
C      Check the convergence
C-----

```

```

C      CALL CHKERR(FORCE1, OLOAD, NB)
C      IF (IEND.EQ.NEPS) GOTO 100
C
C      - adjust the increments,
C      if necessary
C      IF (ITER.GT.25) THEN
C          CALL MULVEC(FORCE, NB)
C          CALL MULVEC(DISP, NB)
C          GAMMA=DSQRT(0.5D0+GAMMA)*0.5D0
C          LAMDA=0.5D0 + GAMMA
C          IF (DINCRLT.1D-4) THEN
C              WRITE (6,*) 'DINCR is ', DINCR
C              STOP
C          ENDIF
C          YSURF=YSURF-DINCR
C          DINCR=0.75D0*DINCR
C          YSURF=YSURF+DINCR
C          GOTO 199
C      ENDIF
C      GOTO 200
C
C      .....
C      write outputs
C      .....
C      100 CONTINUE
C      WRITE (6,*) '-> Writing output data'
C      IF (NSLPSK.EQ.0) CALL OUTPUT(INCR,XU,YU,ZU)
C      IF (NSLPSK.EQ.1) CALL FNDSS(INCR)
C      IF (TOTFY.LT.CF1*YLOAD) THEN
C          WRITE(FILNM, '(A4,I2.2,A4)') 'tim', INCR, 'out'
C          OPEN (18, FILE=FLNM, STATUS='old')
C          CLOSE (18, STATUS='delete')
C          ELSEIF (TOTFY.GT.CF2*YLOAD) THEN
C              GOTO 500
C          ENDIF
C
C      -- update coordinates
C      for new stiffness
C      and save the old loads
C      CALL ASNVEC(X,XU,NP)
C      CALL ASNVEC(Y,YU,NP)
C      CALL ASNVEC(Z,ZU,NP)
C      CALL ADDVEC(LOAD, FORCE, OLOAD, NB)
C      LAMDA=0.5D0 + GAMMA
C      GOTO 80
C
C      500 CONTINUE
C      WRITE (6,*) '*** END OF TTIM ***'
C      WRITE (6,*)
C      close(6)
C      STOP
C      END
C
C      .....
C      SUBROUTINE NONSLP(NP, NE, FXYZ, JOB)

```

```

C      This subroutine detects contact nodes and determine the contact
C      surfaces that the contact nodes belong to. Then it calls subroutines
C      PRECAL and SHRCAL to calculate pressure and shear stresses,
C      respectively. Then it calls subroutine CNVTPS to convert pressure
C      and shear stresses to nodal forces. If output is required (job=0)
C      then it writes output.
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      PARAMETER(NPMAX=1300, NEMAX=600, NNODE=8, NDDOF=3, NNOD=4)
C      COMMON /ELNOD/ LNODE(NEMAX, NNODE)
C      COMMON /COORD3/ XU(NPMAX), YU(NPMAX), ZU(NPMAX)
C      COMMON /SURFAC/ YSURF, YMIN, XBOT, RBAR, THETA0, THETA1, DTHETA
C      COMMON /OTHER/ PI, RATIO, SLIP, ANG, P12, NPSCT
C      COMMON /PRFRM/ TOTFY, THRST, RESIS, ZSNK, KSRG
C      COMMON /SHRTP/ XBOT0, YBOT0, NSHTP, NSLPSK
C
C      REAL*8 FXYZ(*)
C      REAL*8 COORD(NNODE, NDDOF), FP(NNODE, NDDOF), FSH(NNODE, NDDOF),
C      & TRACT(NNODE), SHEAR(NNODE), SAYSHR(NEMAX), SAVP(NEMAX)
C      INTEGER CNTELM(NEMAX), IPS(NEMAX)
C
C      find the bottom point and detect contact elements
C
C      CALL FNDBOT(NP, XU, YU, ZU, XBOT0, YBOT0, ZBOT)
C      CALL DETCNT(NE, CNTELM, NECNT, XU, YU, ZU)
C
C      loop through all the contact elements to calculate
C      normal pressure and shear stress
C
C      IP=0
C      DO 30 I=1, NECNT
C          IE=CNTELM(I)
C          DO 40 J=1, NNOD
C              ID=LNODE(IE, J)
C              -- avoid repeating the same node
C              on different elements
C              DO 35 K=1, IP
C                  IF (ID.EQ.IPS(K)) GOTO 40
C              CONTINUE
C              IF (YU(ID).LE.YSURF) THEN
C                  35
C                  -- calculate normal pressure
C                  CALL PRECAL(CNTELM, NECNT, XU(ID), YU(ID), ZU(ID),
C                  & PRE, ICNT)
C                  ICNT = 1, the node is in contact with terrain
C                  ICNT = 0, the node is NOT in contact with terrain
C              IF (ICNT.EQ.0) GOTO 40
C

```

```

C C
C C -- calculate shear stress
C C PRE : normal pressure
C C TAUNT : internal shear stress
C C TAURUB: rubber-terrain shear stress
C C
C C IP=IP+1
C C IPS(IP)=ID
C C SAVP(IP)=PRE
C C CALL SHRCAL(CNTELM,NECNT,XU(ID),YU(ID),ZU(ID),
C C PRE,TAUNT,TAURUB,DJ)
C C &
C C SAVSHR(IP)=(1-D0-RATIO)*TAUNT + RATIO*TAURUB
C C ENDF
C C 40 CONTINUE
C C 30 CONTINUE
C C
C C -- to convert or output the stresses
C C
C C TOTFY=0.D0
C C THRST=0.D0
C C RESIS=0.D0
C C DO 70 I=1,NECNT
C C IF=CNTELM(I)
C C
C C -- initialize temporary pressure
C C and shear stress arrays
C C
C C IL=0
C C CALL NULVEC(TRACT,NNOOD)
C C CALL NULVEC(SHEAR,NNOOD)
C C DO 50 J=1,NNOOD
C C ID=LNODE(IE,J)
C C DO 55 K=1,IP
C C IF (ID.EQ.IPS(K)) THEN
C C TRACT(J)=SAVP(K)
C C SHEAR(J)=SAVSHR(K)
C C GOTO 52
C C ENDF
C C 55 CONTINUE
C C GOTO 50
C C IL=IL+1
C C 50 CONTINUE
C C
C C -- if no contact point in this element,
C C then skip
C C IF (IL.EQ.0) GOTO 70
C C
C C -- job=1 convert pressure and shear
C C stress to nodal forces
C C and store in vector FXYZ
C C
C C -- job=0 output
C C IF (JOB.EQ.1) THEN
C C CALL ECOORD(IE,COORD,NNOOD,XU,YU,ZU)
C C CALL CNVTPS(COORD,NDOOF,NNOOD,TRACT,SHEAR,F,P,FSH)
C C DO 45 J=1,NNOOD

```

```

RR0=DSQRT(YSURF*YSURF + XSTART*XSTART)
A0=0.25D0*SINK
B0=0.75D0*SINK

C
C This method, developed by Songlao Chen, is a combination of secant,
C false position, bisection and quasi-Newton methods.
C
C Note: A0 < B0, initial range of the solution
C K -----> no. of function calls.
C EPSSKG ----> allowable linkage error.
C MAXF ----> maximum no. of function call
C X -----> solution.
C
C FC=1D99
C FD=1D99
C MAXF=50
C A=A0
C B=B0
C
C Check the function values at given ends, and direct the
C search as appropriate.
C
C KT=0
C CALL SLPSNK(A,FA,KT,INCR,0)
C CALL SLPSNK(B,FB,KT,INCR,0)
C
C If signs are different, then use secant:false position-bisection
C method, starting at 1.
C
C IF (FB*FA LT 0D0) GOTO 1
C
C If not, try to find the range that function changes its signs
C
C 2 CK=(FB-FA)/(B-A)
C C=A-FA/CK
C C=MIN(0.9D0*RR0,MAX(C,1D-10))
C CALL SLPSNK(C,FC,KT,INCR,0)
C
C check convergence.
C
C CALL CNVRGN(EPSSKG,FC,C,A,B,X,IC)
C IF (IC.EQ.1) GOTO 5
C IF (KT.GT.MAXF) CALL ERROR1
C
C TEMP1=FC*FB
C TEMP2=FC*FA
C IF ((TEMP1 LT 0D0) .OR. (TEMP2 LT 0D0)) THEN
C   IF (C LT A) THEN
C     B=A
C     FB=FA
C     A=C
C     FA=FC

```

```

ELSEIF (C.GT.A .AND. C.LT.B) THEN
  IF (TEMP1 LT 0D0) THEN
    A=C
    FA=FC
  ELSE
    B=C
    FB=FC
  ENDIF
ELSEIF (C.GT.B) THEN
  A=B
  FA=FB
  B=C
  FB=FC
ENDIF
C
C If the range has been found, then use secant:false position-bisection
C method, starting at 1.
C
C GOTO 1
ENDIF
C
C No feasible range has been found, conduct quasi-Newton search
C
C TEMPA=DABS(FA)
C TEMPB=DABS(FB)
C TEMPC=DABS(FC)
C IF (TEMPA.GT.TEMPB) THEN
C   A=B
C   FA=FB
C   B=C
C   FB=FC
C ELSE
C   B=A
C   FB=FA
C   A=C
C   FA=FC
C ENDIF
C GOTO 2
C
C 1 CK=(FB-FA)/(B-A)
C C=A-FA/CK
C C=MIN(0.9D0*RR0,MAX(C,1D-10))
C CALL SLPSNK(C,FC,KT,INCR,0)
C
C check convergence.
C
C CALL CNVRGN(EPSSKG,FC,C,A,B,X,IC)
C IF (IC.EQ.1) GOTO 5
C
C use secant:false position method.
C
C IF (FC*FA.GT.0D0) THEN

```



```

C      REAL*8 SHAPE(3)
C      YBOT=1.D+30
DO 5 I=1,NP
  IF (Y(I) LT YBOT) THEN
    YBOT=Y(I)
  IFBOT=I
  ENDF
5 CONTINUE
C      to avoid numerical instability, interpolation is used to locate
C      the new bottom (or minimum) point.
C
C      I1=IBOT+NPSCT
I2=IBOT
I3=IBOT-NPSCCT
TEMP=Y(I1)+Y(I3)-2.D0*Y(I2)
IF (TEMP.NE 0.D0) THEN
  ETA=0.5D0*(Y(I1)-Y(I3))/TEMP
  CALL LINSHR(SHAPE,ETA)
  YBOT=Y(I1)*SHAPE(1)+Y(I2)*SHAPE(2)+Y(I3)*SHAPE(3)
  XBOT=X(I1)*SHAPE(1)+X(I2)*SHAPE(2)+X(I3)*SHAPE(3)
  ZBOT=Z(I1)*SHAPE(1)+Z(I2)*SHAPE(2)+Z(I3)*SHAPE(3)
ELSE
  XBOT=X(I2)
  ZBOT=Z(I2)
ENDIF
C
C      RETURN
C      END
C-----
SUBROUTINE LINSHR(SHAPE,ETA)
C      This subroutine provides shape functions for linear interpolation
C
C      REAL*8 SHAPE(*) ,ETA
C      SHAPE(1)=0.5D0*ETA*(ETA+1.D0)
C      SHAPE(2)=1.D0-ETA*ETA
C      SHAPE(3)=0.5D0*ETA*(ETA+1.D0)
C      RETURN
C      END
C-----
SUBROUTINE SLPSNK(Z0,WORK,KT,INCR,FINAL)
C      This subroutine takes the slip-sinkage effect into consideration,
C      and detects contact nodes and determines the contact
C      surfaces that the contact nodes belong to. Then it calls subroutines
C      PRECAL and SHRCAL to calculate pressure and shear stresses,
C      respectively. Then it calls subroutine CNVTPS to convert pressure
C      and shear stresses to nodal forces. If output is required
C      (FINAL=1) then it writes output.
C
C      IMPLICIT REAL*8 (A-H,O-Z)
PARAMETER (NPMAX=1300,NEMAX=600,NNODE=8,NNO=4,NDOF=3)
COMMON /COORD3/ XU(NPMAX),YU(NPMAX),ZU(NPMAX)
COMMON /ELNOD/ LNODE(NEMAX,NNODE)
COMMON /EINF/ NP,NE,NM,NB
COMMON /SURFAC/ YSURF, YMIN,XBOT,RBAR,THETA0,THETA1,DTHETA
REAL*8 XR(NPMAX),YR(NPMAX),
& SAVP(NEMAX),SAVSHR(NEMAX),SAVZS1(NEMAX),SAVZS2(NEMAX),
& COORD(NNO,NDOF),F,P(NNO,NDOF),FSH(NNO,NDOF),
& TRACT(NNO),SHEAR(NNO),ZS1(NNO),ZS2(NNO)
INTEGER CNTELM(NEMAX),CNTSRF(NEMAX),IPS(NEMAX),FINAL
COMMON /SRCHS/ TH,TO,RR0,SINK,CNTSRF,NECNT0
COMMON /SKGPM/ ALAR,BIBR,PK1,PKREPSSKG
COMMON /PRFRM/ TOTFY,THRST,RESIS,ZSNK,KSRC
C      KT=KT+1
KSRC=KT
WRITE(6,*) '      Searching for slip-sinkage: iteration #',KT
WRITE(6,*) '      Current value is:',Z0
C      GAMMA0=DACOS(-(YSURF+Z0)/RR0)
XL=RR0*(DSIN(GAMMA0)-DSIN(THETA0))
OMEGA=GAMMA0-THETA0
ZSNK=Z0
C      change the coordinates due to the slip-sinkage at wheel center
C
C      DO 2 I=1,NP
C      CALL ROTATE(XU(I),YU(I),OMEGA,XR(I),YR(I))
2 CONTINUE
C      DO 1 I=1,NP
C      CALL TRANSL(XR(I),YR(I),XL,Z0,XR(I),YR(I))
1 CONTINUE
C      find the contact elements
C
C      CALL DETCNT(NE,CNTELM,NECNT,XR,YR,ZU)
C      find the bottom point
C
C      CALL FNDBOT(NP,XR,YR,ZU,XBOT,YBOT,ZBOT)
RRP=DSQRT(XBOT**2+YBOT**2)
THETA=DASIN(XBOT/RRP)
CALL LOCATE(CNTSRF,NECNT0,XG,ZBOT,YG,THETA,3,ALPHA)
XBOT=-YG*DTAN(THETA)
C      loop through all the contact elements to calculate
C      normal pressure and shear stress
C
C      IP=0
DO 30 I=1,NECNT

```



```

C      CALL OUTPUT(INCR,XU,YU,ZU)
C      save the normal pressure and shear stress
C
C      DO 70 I=1,NECNT
C      IE=CNTELM(I)
C      CALL SETMP(IE,FINAL,IP,IPS,SAVP,SAVSHR,SAVZS1,SAVZS2,
C      &      TRACT,SHEAR,ZS1,ZS2,IL)
C
C      - if no contact point in this element,
C      then skip
C
C      IF (IL.EQ.0) GOTO 70
C
C      CALL ECOORD(IE,COORD,NNOD,XR,YR,ZR)
C      WRITE (18,80) IE
C      WRITE (18,81) (LNODE(IE,J),J=1,NNOD)
C      DO 60 K=1,NNOD
C      PRE=TRACT(K)
C      WRITE (18,82) (COORD(K,J)*100 D0,J=1,NDDOF),PRE,SHEAR(K)
C      CONTINUE
C      WRITE (18,82) (COORD(K,J)*100 D0,J=1,NDDOF),PRE,SHEAR(K)
C      CONTINUE
C      CLOSE(18)
C      ENDF
C      80 FORMAT(1X,'Element ',I3)
C      81 FORMAT(1X,4(14,2X))
C      82 FORMAT(1X,3(F12.5,2X),2(G12.5,2X))
C      RETURN
C      END
C-----
C      SUBROUTINE SETMP(IE,FINAL,NIP,IPS,SAVP,SAVSHR,SAVZS1,SAVZS2,
C      &      TRACT,SHEAR,ZS1,ZS2,IL)
C
C      This subroutine initializes temporary arrays for
C      pressure, shear and slip-sinkage
C
C      PARAMETER (NPMAX=1300,NEMAX=600,NNODE=8,NNOD=4)
C      IMPLICIT REAL*8 (A-H,O-Z)
C      COMMON /ELNOD/ LNODE(NEMAX,NNODE)
C      REAL*8 SAVP(1),SAVSHR(1),SAVZS1(1),SAVZS2(1),
C      &      TRACT(1),SHEAR(1),ZS1(1),ZS2(1)
C      INTEGER IPS(1),FINAL
C
C      IL=0
C      CALL NULVEC(TRACT,NNOD)
C      IF (FINAL.EQ.1) THEN
C      CALL NULVEC(SHEAR,NNOD)
C      ELSE
C      CALL NULVEC(ZS1,NNOD)
C      CALL NULVEC(ZS2,NNOD)
C      ENDIF
C
C      DO 50 J=1,NNOD
C
ID=LNODE(IE,J)
DO 55 K=1,NIP
IF (ID.EQ.IPS(K)) THEN
TRACT(J)=SAVP(K)
IF (FINAL.EQ.1) THEN
SHEAR(J)=SAVSHR(K)
ELSE
ZS1(J)=SAVZS1(K)
ZS2(J)=SAVZS2(K)
ENDIF
GOTO 52
ENDIF
55 CONTINUE
GOTO 50
52 IL=IL+1
50 CONTINUE
RETURN
END
C-----
C      SUBROUTINE EQWRK(NNOD,NDDOF,COORD,TRACT,ZS1,ZS2,WRKDN)
C
C      This subroutine calculates the difference in equivalent work done
C      by forces along both predicted slip-sinkage and geometrically
C      allowable slip-sinkage.
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      PARAMETER (LCCOD=2,IDER2=2)
C      COMMON /COEFFS/ COEFF(10,3),SAMP(2,2),NYCF(3),NGP
C
C      REAL*8 COORD(NNOD,3),TRACT(1),ZS1(1),ZS2(1)
C      REAL*8 DERIV(2,3),DER2(2,4),FUN2(4)
C
C      WRKDN=0.D0
C      DO 10 J=1,NGP
C      DO 20 K=1,NGP
C      XL=SAMP(J,1)
C      ETA=SAMP(K,1)
C      CALL FMLIN2(3,DER2,IDER2,FUN2,XI,ETA)
C      ANYP=0.D0
C      ANYZS=0.D0
C      DO 5 I=1,NNOD
C      ANYP=ANYP + FUN2(I)*TRACT(I)
C      ANYZS=ANYZS + FUN2(I)*(ZS1(I)-ZS2(I))
C      CONTINUE
C      CALL NULL(DERIV,LCCOD,LCCOD,NDDOF)
C      CALL MATMUL(DER2,COORD,DERIV,LCCOD,NNOD,NDDOF)
C      PRD=SAMP(J,2)*SAMP(K,2)
C      AX=DERIV(1,2)*DERIV(2,3)-DERIV(1,3)*DERIV(2,2)
C      AY=DERIV(1,3)*DERIV(2,1)-DERIV(1,1)*DERIV(2,3)
C      AZ=DERIV(1,1)*DERIV(2,2)-DERIV(1,2)*DERIV(2,1)
C
C      ANOM=DSQRT(AX*AX + A*AY + AZ*AZ)

```

```

C
C      -- neglect the AZ*AZ term
C      because slip-sinkage occurs
C      in X-Y plane.
C      ANOM=DSORT(AX*AX+AY*AY)
      WRKON=WRKON + ANYP*ANYZS*ANGC*PRD
20  CONTINUE
10  CONTINUE
      RETURN
      END
C-----
SUBROUTINE UPDATE(INCR,DISPT,FORCE,DISPT,INCR)
C
C      This subroutine saves previous iterational loads,
C      updates the coordinates,
C      calculates new loads and
C      updates Young's moduli.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(NPMAX=1300,NEMAX=600,NNODE=8,NDDOF=2,NMMAX=3)
      COMMON /COORD3/ X(NPMAX),Y(NPMAX),Z(NPMAX)
      COMMON /COORD3/ XU(NPMAX),YU(NPMAX),ZU(NPMAX)
      COMMON /BOUND/ NBC(NPMAX,NDDOF)
      COMMON /MEINF/ NP,NE,NM,NB
      COMMON /MATER/ YOUNG(NPMAX),POISS(NMMAX),NMAT(NEMAX)
      COMMON /SURF/ YSURF,Y1,YIN,XBOT,RBAR,THETA0,THETA1,DTHETA
      COMMON /OTHER/ PI,RATIO,SLIP,ANG,PI2,NPSC1
      COMMON /SHRTP/ XBOTQ,YBOTQ,NSHTP,NSLPSK
      COMMON /PRFRM/ TOTFY,THRS*,RESIS,ZSNK,KSRC
      REAL*8      LAMDA
      COMMON /SRCH/ ERR(3),EPS,LAMDA,GAMMA,CRTREF,
&      NEPS,END,ITER,NITER
C
      REAL*8  FXYZ(NPMAX,NDDOF)
      REAL*8  DISPT(*),FORCE(*),FORCE1(*)
C
C      -- save old loads
C
      CALL ASNVEC(FORCE1,FORCE,NB)
C
C      -- update coordinates
C      -- find YMIN and XBOT
C
      DO 1 I=1,NP
      DO 2 K=1,NDDOF
      L=NBC(I,K)
      IF (L.EQ.0) THEN
      XX=0.D0
      ELSE
      XX=DISPT(L)
      IF (K.EQ.1) THEN
      XU(I)=XX+Y(I)

```

```

      ELSEIF (K.EQ.2) THEN
      YU(I)=XX+Y(I)
      ELSE
      ZU(I)=XX+Z(I)
      ENDIF
2  CONTINUE
1  CONTINUE
C
      CALL FNDBOT(NP,XU,YU,ZU,XBOT,YMIN,ZBOT)
C
C      -- update RBAR
C
      IF (ITER.EQ.2) THEN
      DFLN=0.D0
      DO 11 I=1,NP
      L=NBC(I,2)
      IF (L.EQ.0) THEN
      XX=0.D0
      ELSE
      XX=DISPT(L)
      ENDIF
      IF (DABS(XX).GT.DFLN) DFLN=DABS(XX)
11  CONTINUE
      CRTREF=(DBLE(NITER)*DFLN/DINCR)**2
      ELSEIF (ITER.GT.2) THEN
      TEMP1=DMAX1(0.001*D0,DMIN1(DBLE(ITER*ITER)/CRTREF,0.999D0))
      LAMDA=0.5D0 + GAMMA*(1.D0 - TEMP1)
      ENDIF
C
C      -- calculate new loads
C
      WRITE (6,*) '--> Calculating loads'
      CALL NULVEC(FXYZ,NP,NDDOF)
      CALL NULVEC(FORCE,NB)
      CALL NONSLP(NP,NE,FXYZ,1)
      DO 55 I=1,NP
      DO 60 K=1,NDDOF
      L=NBC(I,K)
      IF (L.EQ.0) GOTO 60
      L1=(I-1)*NDDOF+K
      FORCE(L)=FXYZ(L1)
60  CONTINUE
55  CONTINUE
C      -- update Young's moduli
C
      CALL FYOUNG(TOTFY)
C
      RETURN
      END
C-----
SUBROUTINE OUTPUT(INCR,XU,YU,ZU)
C
C      This subroutine writes the tractive performance into file

```

```

C      TTIM??OUT, where ?? is the number of increment, and
C      the coordinates
C
C      PARAMETER (NPMAX=1300,NEMAX=600,NDDOF=3)
C      IMPLICIT REAL*8 (A-H,O-Z)
C      COMMON /AEINF/ NP,NE,NM,NB
C      COMMON /PRFRM/ TOTFY,THRST,RESIS,ZSNK,KSRC
C      COMMON /SURFAC/ YSURF,YMIN,XBOT,RBAR,THETA0,THETA1,DTHETA
C      COMMON /SHRTP/ XBOT0,YBOT0,NSHTP,NSLPSK
C
C      REAL*8 FXYZ(NPMAX*NDDOF),XU(?),YU(?),ZU(?)
C      CHARACTER FLNM*12
C
C      WRITE(FLNM,(A4,I2,2,A4)) 'tim', INCR, 'out'
C      OPEN (18,FILE=FLNM)
C      WRITE (18,') ' Normal reaction   thrust   resistance  RBAR'
C      WRITE (18,51) TOTFY, THRST, RESIS, RBAR
C      WRITE (18,') ' Sinkage   YSURF   slip-sinkage'
C      WRITE (18,52) (YSURF-YMIN),YSURF,ZSNK,KSRC
C      WRITE (18,') ' New Coordinates for increment ,INCR
C
C      DO 4 I=1,NP
C      TEMPX=XU(I)*100.D0
C      TEMPY=YU(I)*100.D0
C      TEMPZ=ZU(I)*100.D0
C      WRITE (18,50) I,TEMPX,TEMPY,TEMPZ
C
C      4 CONTINUE
C      WRITE (18,') ' Pressure and Shear Stress Distributions'
C      IF (NSLPSK.EQ.0) THEN
C      CALL NONSLP(NP,NE,FXYZ,0)
C      CLOSE(18)
C      ENDIF
C      50 FORMAT(5X,I4,3(F12.5,2X))
C      51 FORMAT(4(2X,F12.5))
C      52 FORMAT(3(2X,F12.5),I5)
C      RETURN
C      END
C

```



```

      BEE(2,K)=DERIV(2,M)
      BEE(4,L)=DERIV(2,M)
      BEE(5,N)=DERIV(2,M)
      BEE(3,N)=DERIV(3,M)
      BEE(5,K)=DERIV(3,M)
      BEE(6,L)=DERIV(3,M)
      1 CONTINUE
      RETURN
      END
C-----
SUBROUTINE ASNVEC(A,B,N)
C
C   This subroutine assigns the vector B to vector A
C
      REAL*8 A(*),B(*)
      DO 1 I=1,N
        A(I)=B(I)
      1 CONTINUE
      RETURN
      END
C-----
SUBROUTINE ADDVEC(A,B,C,N)
C
C   This subroutine adds vector B to vector A,
C   and assigns it to vector C
C
      REAL*8 A(*),B(*),C(*)
      DO 1 I=1,N
        C(I)=A(I)+B(I)
      1 CONTINUE
      RETURN
      END
C-----
SUBROUTINE SUBVEC(A,B,C,N)
C
C   This subroutine subtracts vector B from vector A,
C   and assigns it to vector C.
C
      REAL*8 A(*),B(*),C(*)
      DO 1 I=1,N
        C(I)=A(I)-B(I)
      1 CONTINUE
      RETURN
      END
C-----
SUBROUTINE FMLIN3(DER,IDER,I,J,K)
C
C   This subroutine calculates gradient matrix of shape functions.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /COEFFS/ COEFF(10,3),SAMP(2,2),NYCF(3),NGP

```

```

      REAL*8 DER(IDER,*)
      ETA =SAMP(K,1)
      XI =SAMP(J,1)
      ZETA=SAMP(L,1)
      ETAM =1 DO-ETA
      XIM =1 DO-XI
      ZETAM=1 DO-ZETA
      ETAP =1 DO+ETA
      XIP =1 DO+XI
      ZETAP=1 DO+ZETA
      -- Derivative of gzu
      DER(1,1)=-0.125DO*ETAM*ZETAP
      DER(1,2)=-0.125DO*ETAP*ZETAP
      DER(1,3)= 0.125DO*ETAP*ZETAP
      DER(1,4)= 0.125DO*ETAM*ZETAP
      DER(1,5)=-0.125DO*ETAM*ZETAP
      DER(1,6)=-0.125DO*ETAP*ZETAP
      DER(1,7)= 0.125DO*ETAP*ZETAP
      DER(1,8)= 0.125DO*ETAM*ZETAP
      -- Derivative of eta
      DER(2,1)=-0.125DO*XIM*ZETAP
      DER(2,2)= 0.125DO*XIM*ZETAP
      DER(2,3)= 0.125DO*XIP*ZETAP
      DER(2,4)=-0.125DO*XIP*ZETAP
      DER(2,5)=-0.125DO*XIM*ZETAP
      DER(2,6)= 0.125DO*XIM*ZETAP
      DER(2,7)= 0.125DO*XIP*ZETAP
      DER(2,8)=-0.125DO*XIP*ZETAP
      -- Derivative of zeta
      DER(3,1)= 0.125DO*XIM*ETAM
      DER(3,2)= 0.125DO*XIM*ETAP
      DER(3,3)= 0.125DO*XIP*ETAP
      DER(3,4)= 0.125DO*XIP*ETAM
      DER(3,5)=-0.125DO*XIM*ETAM
      DER(3,6)= 0.125DO*XIM*ETAP
      DER(3,7)=-0.125DO*XIP*ETAP
      DER(3,8)=-0.125DO*XIP*ETAM
      RETURN
      END
C-----
SUBROUTINE SAVEKM(IE,EKM,IKM,EKMSAV,NEMAX)
C
C   This subroutine saves the element stiffness matrices in EKMSAV
C
      REAL*8 EKM(IKM,*),EKMSAV(NEMAX,*)
      L=0
      DO 10 I=1,IKM
        DO 10 J=1,IKM
          L=L+1
          EKMSAV(IE,L)=EKM(I,J)

```

```

10 CONTINUE
RETURN
END
C-----
SUBROUTINE RESEKM(IE,EKM,IKM,EKMSAV,NEMAX)
C
C This subroutine restores the element stiffness matrices
C from EKMSAV
C
REAL*8 EKM(IKM,*),EKMSAV(NEMAX,*)
C
L=0
DO 10 I=1,IKM
DO 10 J=1,IKM
L=L+1
EKM(I,J)=EKMSAV(IE,L)
IF (I.NE.J) EKM(J,I)=EKM(I,J)
10 CONTINUE
RETURN
END
C-----
SUBROUTINE MSMULT(A,I,A,C,M,N)
C
C This subroutine multiplies matrix A(M,N) by constant C
C
REAL*8 A(IA,*),C
C
DO 10 I=1,M
DO 10 J=1,N
IF (A(I,J).NE.0) A(I,J)=C*A(I,J)
10 CONTINUE
RETURN
END
C-----
SUBROUTINE TREEX3(XJAC,UAC,XJAC1,UJAC1,DET,IE)
C
C This subroutine calculates the Jacobian determinant |J|
C and the inverse of [J]
C
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 XJAC(UAC,*),XJAC1(UJAC1,*)
C
DET= XJAC(1,1)*XJAC(2,2)*XJAC(3,3)-XJAC(3,2)*XJAC(2,3)
DET=DET-XJAC(1,2)*XJAC(2,1)*XJAC(3,3)+XJAC(3,1)*XJAC(2,3)
DET=DET+XJAC(1,3)*XJAC(2,1)*XJAC(3,2)-XJAC(3,1)*XJAC(2,2)
IF (DET.LT.1D-30) THEN
WRITE(6,900) DET,IE
900 FORMAT('TTM ERROR Jacobian =E15.7, <= 0 at ',I4)
STOP
ENDIF
XJAC1(1,1)= XJAC(2,2)*XJAC(3,3)-XJAC(3,2)*XJAC(2,3))DET
XJAC1(2,1)=(-XJAC(2,1)*XJAC(3,3)+XJAC(3,1)*XJAC(2,3))DET
XJAC1(3,1)=(-XJAC(3,1)*XJAC(2,2)+XJAC(2,1)*XJAC(3,2))DET
XJAC1(1,2)=(-XJAC(1,2)*XJAC(3,3)+XJAC(3,2)*XJAC(2,3))DET
XJAC1(2,2)=(-XJAC(2,2)*XJAC(3,3)+XJAC(3,1)*XJAC(2,3))DET
XJAC1(3,2)=(-XJAC(3,2)*XJAC(2,3)+XJAC(2,1)*XJAC(3,3))DET
XJAC1(1,3)=(-XJAC(1,3)*XJAC(2,3)+XJAC(2,2)*XJAC(3,3))DET
XJAC1(2,3)=(-XJAC(2,3)*XJAC(3,3)+XJAC(3,1)*XJAC(2,3))DET
XJAC1(3,3)=(-XJAC(3,3)*XJAC(2,2)+XJAC(2,1)*XJAC(3,2))DET
RETURN
END
C-----
SUBROUTINE MATADD(A,IA,B,IB,M,N)
C
C This subroutine performs matrix addition C[i]=A[i]+B[i]
C
REAL*8 A(IA,*),B(IB,*),C
C
DO 1 I=1,M
DO 1 J=1,N
A(I,J)=A(I,J)+B(I,J)
1 CONTINUE
RETURN
END
C-----
SUBROUTINE SPARIN(A,N,KDIAG)
C
C This subroutine is Gaussian elimination using Sky line method
C
REAL*8 A(*),X
INTEGER KDIAG(*)
C
A(1)=DSORT(A(1))
DO 1 I=2,N
KI=KDIAG(I-1)-1
L=KDIAG(I-1)-KI+1
DO 2 J=L,I
X=A(KI~J)
A(KI~J)=A(KI~J)-X*A(KI~I)
KJ=KDIAG(J)-J
IF (J.EQ.1) GOTO 2
LBAR=KDIAG(J-1)-KJ+1
LBAR=MAX0(L,LBAR)
IF (LBAR.EQ.J) GOTO 2
M=J-1
DO 3 K=LBAR,M
X=X-A(KI~K)*A(KJ~K)
3 X=X-A(KI~K)*A(KJ~K)
2 A(KI~J)=X/A(KJ~J)
1 A(KI~I)=DSORT(X)
RETURN
END
C-----
SUBROUTINE SPABAC(A,B,N,KDIAG)
C
C Back-substitution

```



```

C
C      This subroutine finds the maximum diagonal of the stiffness matrix.
C
C      INTEGER KDIAG(*) ,IG(NEMAX,*)
C
C      DO 1 I=1, IDOF
C        IWP1=1
C        IF (IG(IE,I) EQ.0) GOTO 1
C        DO 2 J=1, IDOF
C          IF (IG(IE,J) EQ.0) GOTO 2
C          IM=IG(IE,I)-IG(IE,J)+1
C          IF (IM.GT. IWP1) IWP1=IM
C        2 CONTINUE
C        K=IG(IE,I)
C        IF (IWP1.GT. KDIAG(K)) KDIAG(K)=IWP1
C      1 CONTINUE
C      RETURN
C      END
C-----
C      SUBROUTINE FYOUNG(TOTFY)
C
C      This subroutine calculates the Young's moduli for a given load
C      using the pre-processed polynomials
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      PARAMETER(NEMAX=600,NMMAX=3)
C      COMMON /COEFFS/ COEFF(10,3),SAMPI(2,2),NYCF(3),NGP
C      COMMON /MATER/ YOUNG(NMMAX),POISS(NMMAX),NMAT(NEMAX)
C      COMMON /MEINF/ NP,NE,NM,NB
C
C      DO 1 I=1,NM
C        YOUNG(I)=0.D0
C        DO 2 J=NYCF(I),1,-1
C          YOUNG(I)=COEFF(J,I) + TOTFY*YOUNG(I)
C        2 CONTINUE
C      1 CONTINUE
C      RETURN
C      END
C-----
C      SUBROUTINE CHKERR(FORCE,FORCE1,OLOA,J,NB)
C
C      This subroutine checks the convergence
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      REAL*8 LAMDA
C      COMMON /SRCH/ ERR(3),EPS,LAMDA,GAMMA,CRTREF,
C      & NEPS,IEND,ITER,NITER
C      REAL*8 FORCE(*),FORCE1(*),OLOAD(*)
C
C      WRITE (6,*) ' -> Checking convergence'
C      ITER=ITER+1
C      IF (ITER.GT.NITER) THEN

```

```

C
C      REAL*8 A(*),B(*),X
C      INTEGER KDIAG(*)
C
C      B(1)=B(1)/A(1)
C      DO 1 I=2,N
C        KI=KDIAG(I)-1
C        L=KDIAG(I-1)-KI+1
C        X=B(I)
C        IF (L EQ.1) GOTO 1
C        M=L-1
C        DO 2 J=L,M
C          X=X-A(KI+J)*B(J)
C        1 B(I)=X/A(KI+1)
C
C      DO 3 IT=2,N
C        I=N+2-IT
C        KI=KDIAG(I)-1
C        X=B(I)/A(KI+1)
C        B(I)=X
C        L=KDIAG(I-1)-KI+1
C        IF (L EQ.1) GOTO 3
C        M=L-1
C        DO 4 K=L,M
C          B(K)=B(K)-X*A(KI+K)
C        3 CONTINUE
C        B(1)=B(1)/A(1)
C      RETURN
C      END
C-----
C      SUBROUTINE FSPARV(TKV,EKM,IKM,IG,KDIAG, IDOF, IE, NEMAX)
C
C      This subroutine assembles the global stiffness matrix.
C
C      REAL*8 TKV(*),EKM(IKM,*)
C      INTEGER KDIAG(*),IG(NEMAX,*)
C
C      DO 1 I=1, IDOF
C        K=IG(IE,I)
C        IF (K EQ.0) GOTO 1
C        DO 2 J=1, IDOF
C          IF (IG(IE,J) EQ.0) GOTO 2
C          IW=K-IG(IE,J)
C          IF (IW.LT.0) GOTO 2
C          IVAL=KDIAG(K)-IW
C          TKV(IVAL)=TKV(IVAL)+EKM(I,J)
C        2 CONTINUE
C      1 CONTINUE
C      RETURN
C      END
C-----
C      SUBROUTINE FKDIAG(KDIAG,IG, IDOF, IE, NEMAX)

```

```

WRITE (6,2) NITER
2 FORMAT(1X,'More than 13, iterations have been conducted',
& 'without convergence.',1X,'--- STOP ---')
STOP
ENDIF
XX=0.D0
YY=0.D0
DO 1 I=1,NB
IF ((FORCE(I),NE.0.D0).OR. (FORCE(I),NE.0.D0)) THEN
XX=XX + (FORCE(I)-FORCE(I))**2
YY=YY + (FORCE(I)+CLOAD(I))**2
ENDIF
1 CONTINUE
IF (YY.GT.0.D0) ERROR=DSORT(XX,YY)
WRITE (6,1) ' .. ITERATION #,ITER
WRITE (6,11)
11 FORMAT(1X,
& ' .. absolute',
& ' .. reference',
& ' .. relative')
WRITE (6,10) XX,YY,ERROR
10 FORMAT(1X,' .. ERRORS =',3(G12.5,2X))
IF (ERROR.LT.EPS) THEN
IEND=IEND+1
ELSE
IEND=0
ENDIF
RETURN
END

C-----
SUBROUTINE CHKCONT(Y,DISPT,NP,ICNT)
C
C This subroutine checks whether the tire is in contact with
C the terrain or not
C
IMPLICIT REAL*8 (A-H,O-Z)
PARAMETER(NPMAX=1300,NDOF=3)
COMMON /BOUND/ NBC(NPMAX,NDOF)
COMMON /SURFAC/ YSURF,YMIN,XBOT,RBAR,THETA0,THETA1,DTHETA

C
C REAL*8 Y('),DISPT(')
C
ICNT=0
DO 1 I=1,NP
L=NBC(I,2)
IF (L.EQ.0) GOTO 1
YY=DISPT(I)+Y(I)
IF (YY.LT.YSURF) THEN
ICNT=1
RETURN
ENDIF
1 CONTINUE
RETURN
END

C-----
SUBROUTINE DATAN(YLOAD,DINCR,CF1,CF2)
C
C This subroutine reads input data
C
IMPLICIT REAL*8 (A-H,O-Z)
CHARACTER*40 ITITLE
PARAMETER(NPMAX=1300,NEMAX=600,NNODE=8,NDOF=3,NMMAX=3)
COMMON /COORD/ X(NPMAX),Y(NPMAX),Z(NPMAX)
COMMON /ELNOD/ LNODE(NEMAX,NNODE)
COMMON /BOUND/ NBC(NPMAX,NDOF)
COMMON /WATER/ YOUNG(NMMAX),POISS(NMMAX),NMAT(NEMAX)
COMMON /MEINF/ NP,NE,NM,NB

C
REAL*8 KC,KPHI,NIN
COMMON /TERAIN/ KC,KPHI,NIN,BB,A0,AU
COMMON /COEFFS/ COEFF(10,3),SAMP(2,2),NYCF(3),NGP
REAL*8 KINT,KRUB
COMMON /SHRINF/ CINT,TANFI,KINT,CRUB,TANFIR,KRUB
COMMON /SURFAC/ YSURF,YMIN,XBOT,RBAR,THETA0,THETA1,DTHETA
COMMON /OTHER/ PI,RATIO,SLIP,ANG,PI2,NPSCT
COMMON /SHRTP/ XBOT,YBOT,XSHTP,NSLPSK
COMMON /SKGPM/ A1AR,B1BR,PK1,PKR,EPSSKG
REAL*8 LAMDA
COMMON /SRCH/ ERR(3),EPS,LAMDA,GAMMA,CRTREF,
& NEPS,IEND,ITER,NITER

C
PI=4.D0/DATAN(1.D0)
D2R=PI/180.D0
PERCT=0.01D0
OPEN (9,FILE='gene3d.in')
READ (9, '(A40)') ITITLE
READ (9,') PRESS,FRCFAC
READ (9,') NP,NE,IP1

C
C read coordinates and boundary conditions
C
DO 10 I=1,NP
READ(9,') IDUMMY,X(I),Y(I),Z(I),(NBC(I,K),K=1,NDOF)
10 CONTINUE

C
C --- convert cm to m
C
CALL MULVEC(X,X,PERCT,NP)
CALL MULVEC(Y,Y,PERCT,NP)
CALL MULVEC(Z,Z,PERCT,NP)

C
C read node numbers and material properties
C
DO 20 I=1,NE
READ (9,') IDUMMY,(LNODE(I,K),K=1,NNODE),NMAT(I)
20 CONTINUE
READ (9,') NM

```

```

DC 30 L=1,NM
  READ (9,'YOUNG(I),POISS(I)
30 CONTINUE
  CLOSE(9)
C
C  read terrain properties in file TERRAIN.DAT
C
C  OPEN (19,FILE='terrain.dat')
C
C  read pressure-sinkage data
C
C  READ (19,'KC,KPHI,NN,BB
BB=BB*PERCT
  READ (19,'A0,AU
  READ (19,'YSURF,DINCR,YLOAD
C
C  read polynomials for Young's moduli, E_1 and E_s
C
C  READ (19,'NYCF(1)
  READ (19,'(COEFF(J,1),NYCF(1),1,1)
  READ (19,'NYCF(2)
  READ (19,'(COEFF(J,2),NYCF(2),1,1)
C
C  read shear parameters
C
C  READ (19,'CINT,PHI,KINT
  TANFIL=DTAN(PHI*D2R)
  KINT=KINT*PERCT
  READ (19,'CRUB,PHI,KRUB
  TANFIL=DTAN(PHI*D2R)
  KRUB=KRUB*PERCT
  READ (19,'ANG,SLIP,RATIO
  ANG=ANG*D2R
  SLIP=SLIP*PERCT
C
C  read slip-sinkage parameters and those for search routines
C
C  READ (19,'A1,B1,PK1
  READ (19,'AR,BR,PKR
  READ (19,'NSLP,SK,EPSSKG
  READ (19,'LAMDA,EPS,NEPS,NITER
  GAMMA=LAMDA - 0.5D0
  READ (19,'NPSCT,NSHTP
  READ (19,'CF1,CF2
  CLOSE(19)
C
C  RETURN
  END

```



```

C      -- after the dead center.
C      calculate new theta1 for
C      the bottom point of Z0
C
C      CALL THENPT(XBOT0,YBOT0,NPT)
C      CALL CALC(JCNTELM,NECNT,NPT,Z0,DDJ,VALI,VALR)
C
C      -- calculate shear/tanghi for
C      the bottom point
C
C      SHRI=DEXP(-DJ0/KINT)*VALI/KINT
C      SHRR=DEXP(-DJ0/KRUB)*VALR/KRUB
C
C      -- using XBOT and Z0 to locate YBOT
C      for calculating bottom pressure
C
C      CALL LOCATE(CNTELM,NECNT,XBOT0,Z0,YBOT,THETA,2,ALPHA)
C      ZSINK=YSURF-YBOT
C      IF (ZSINK.GT.1D-15) THEN
C        PBOT=(KC/BB+KPHI)*ZSINK**NN
C      ELSE
C        WRITE(6,*) 'trouble !! sinkage is less than zero'
C        WRITE(6,*) YSURF,YBOT,ZSINK
C        STOP
C      ENDIF
C
C      -- calculate the equivalent shear
C      displacements
C
C      DJEQI=KINT*DLOG(1D0-SHRI/PBOT)
C      DJEQR=KRUB*DLOG(1D0-SHRR/PBOT)
C
C      -- reset theta0 and theta1 to
C      calculate the shear displacement
C      from the bottom point
C      to the current point
C
C      THETA0=THETA1
C      CALL THENPT(X0,Y0,NPT)
C      CALL CALC(JCNTELM,NECNT,NFT,Z0,DDJ,VALI,VALR)
C
C      -- calculate the shear stresses
C      using the equivalent disps.
C
C      TAU1=PRE*TANFIR*(1D0-DEXP(-(DJEQI+DDJ)/KINT))
C      TAU2=PRE*TANFIR*(1D0-DEXP(-(DJEQR+DDJ)/KRUB))
C      TAU1NT=TAU1NT + TAU1
C      TAU2R=TAU2R + TAU2
C      ENDIF
C      ENDIF
C      RETURN
C      END
C
C      SUBROUTINE THENPT(X0,Y0,NPT)
C
C      C      This subroutine calculates the current contact angle and
C      C      the no. of divisions needed to obtain the shear displacement j.
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      COMMON /SURFAC/ YSURF,YMIN,XBOT,RBAR,THETA0,THETA1,DTHETA
C      COMMON /OTHER/ PI,RATIO,SLIP,ANG,PI2,NPSC1
C
C      THETA1=DATAN(-X0/Y0)
C      NPT=IDNINT((THETA0-THETA1)/ANG)
C      IF (NPT.EQ.0) THEN
C        NPT=1
C      DTHETA=THETA0-THETA1
C      ELSE
C        DTHETA=(THETA0-THETA1)/DBLE(NPT)
C      ENDIF
C      RETURN
C      END
C
C      SUBROUTINE CALC(JCNTELM,NECNT,NPT,Z0,DISP,VALI,VALR)
C
C      C      This subroutine calculates shear displacement
C      C      using Gauss quadrature.
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      REAL*8 KINT,KRUB
C      COMMON /SHRINF/ CINT TANFIL,KINT,CRUB,TANFIR,KRUB
C      COMMON /COEFFS/ COEFF(10,3),SAMP(2,2),NYCF(3),NGP
C      COMMON /SURFAC/ YSURF,YMIN,XBOT,RBAR,THETA0,THETA1,DTHETA
C      COMMON /OTHER/ PI,RATIO,SLIP,ANG,PI2,NPSC1
C      COMMON /SHRTP/ XBOT0,YBOT0,NSHTP,NSLPSK
C
C      INTEGER CNTELM(*)
C
C      THE1=THETA0
C      SUMINT=0D0
C      SUMYI=0D0
C      SUMVR=0D0
C      DJACB=0.5D0*DTHETA
C      DO 1 I=1,NPT
C        THE2=THETA0-DTHETA*DBLE(I)
C        IF (I.EQ.NPT) THE2=THETA1
C        SUM=0D0
C        SUMI=0D0
C        SUMR=0D0
C        DO 2 J=1,NGP
C          THETA=0.5D0*(THE1+THE2 + (THE2-THE1)*SAMP(J,1))
C          CALL LOCATE(CNTELM,NECNT,XG,Z0,YG,THETA,3,ALPHA)
C          R=YG/DCOS(THETA)
C          FUNC=R*DCOS(THETA-ALPHA)-RBAR*(1D0-SLIP)*DCOS(ALPHA)

```

```

SUM=SUM+FUNC*SAMP(J,2)
IF (NSHTPEQ.1) THEN
  XG=YG*DTAN(THETA)
  CALL PRECAL(CNTELM,NECNT,XG,YG,Z0,PRE,ICNT)
  assuming that within the increment (THE2-THE1)
  the contact shape is circular.
  RTMP=R*(THE1-THE2)/(1.D0-SLIP)*(DSIN(THE1)-DSIN(THE2)))
  DJ=SUMINT + RTMP
  SUML=SUMI + PRE*DEXP(DJ/KINT)*FUNC*SAMP(J,2)
  SUMR=SUMR + PRE*DEXP(DJ/KRUB)*FUNC*SAMP(J,2)
  ENDIF
2 CONTINUE
  SUMINT=SUMINT + SUM*DJACB
  IF (NSHTPEQ.1) THEN
    SUMVI=SUMVI + SUMI*DJACB
    SUMVR=SUMVR + SUMR*DJACB
  ENDIF
  THE1=THE2
1 CONTINUE
  DISP=SUMINT
  VALI=SUMVI
  VALR=SUMVR
  RETURN
END

SUBROUTINE LOCATE(CNTELM,NECNT,Y0,Z0,CORD,THETA,LCT,ALPHA)
C
C This subroutine locates the contact element in which the given
C point resides, and completes the coordinates of the given point.
C
C LCT = 1 given Y and Z, search for X (Y,Z,X,THETA,1)
C LCT = 2 given X and Z, search for Y (X,Z,Y,THETA,2)
C LCT = 3 given Y and Z and theta, search for X and dY/dX (X,Z,Y,THETA,3)
C
IMPLICIT REAL*8 (A-H,O-Z)
PARAMETER (NPMAX=1300,NEMAX=600,NNODE=8,NNOD=4)
COMMON /COORD3/ XU(NPMAX),YU(NPMAX),ZU(NPMAX)
C
REAL*8 A(NNOD),B(NNOD),C(NNOD)
INTEGER CNTELM(1)
C
IANSFD=0
Y1=Y0
Z1=Z0
DO 1 I=1,NECNT
  IE=CNTELM(I)
  CALL CALABC(IE,ZU,C)
  IF (LCT.EQ.1) THEN
    CALL CALABC(IE,YU,B)
  ELSEIF (LCT.EQ.2) THEN
    CALL CALABC(IE,XU,B)
  ELSE
    CALL CALABC(IE,XU,B)
  ENDIF
  X1=(Y1-B(1)+B(3)*ETA)/(B(2)+B(4)*ETA)
  CALL ROUND(X1)
  IF (DABS(X1) LE 1.D0) THEN
    IANSFD=1
    GOTO 2
  ENDIF
1 CONTINUE
2 CONTINUE
  IF (IANSFD.EQ.0) THEN
    WRITE (6,*) 'no solution of xi and eta in LOCATE'
    STOP
  ENDIF
C
C the element has been located, then completes the task

```

```

CALL CALABC(IE,XU,B)
ELSE
  CALL CALABC(IE,XU,A)
  CALL CALABC(IE,YU,B)
  TANTHE=DTAN(THETA)
  DO 3 J=1,NNOD
    B(J)=B(J)*TANTHE+A(J)
  CONTINUE
  Y1=0.D0
  ENDIF
3
C
C -- prepare the coefficients of
C the quadratic equation of eta.
C
P=B(4)*C(3)-C(4)*B(3)
Q=B(4)*C(1)-C(4)*B(1)+B(2)*C(3)-C(2)*B(3)+Y1*C(4)-Z1*B(4)
R=B(2)*C(1)-C(2)*B(1)+Y1*C(2)-Z1*B(2)
DELTA=Q*Q-4.D0*P*R
C
C -- if delta < 0, no solution, go for
C next element.
C
IF (DELTA LT 0.D0) GOTO 1
ETA1=(-Q+DSQRT(DELTA))/(2.D0*P)
ETA2=(-Q-DSQRT(DELTA))/(2.D0*P)
CALL ROUND(ETA1)
CALL ROUND(ETA2)
IF (DABS(ETA1) LE 1.D0) THEN
  ETA=ETA1
ELSEIF (DABS(ETA2) LE 1.D0) THEN
  ETA=ETA2
ELSE
  -- if not (-1 <= eta <= 1), no solution,
  go for next element.
  GOTO 1
ENDIF
X1=(Y1-B(1)+B(3)*ETA)/(B(2)+B(4)*ETA)
CALL ROUND(X1)
IF (DABS(X1) LE 1.D0) THEN
  IANSFD=1
  GOTO 2
ENDIF
1 CONTINUE
2 CONTINUE
  IF (IANSFD.EQ.0) THEN
    WRITE (6,*) 'no solution of xi and eta in LOCATE'
    STOP
  ENDIF
C
C the element has been located, then completes the task

```

```

C      IF (LCT.EQ.1) THEN
C      CALL CALXYZ(XI,ETA,NNOD,IE,XU,CORD)
C      ELSEIF (LCT.EQ.2) THEN
C      CALL CALXYZ(XI,ETA,NNOD,IE,YU,CORD)
C      ELSE
C      CALL CALXYZ(XI,ETA,NNOD,IE,YU,CORD)
C      CALL CALALF(XI,ETA,NNOD,IE,ALPHA)
C      ENDIF
C      RETURN
C      END
C-----
C      SUBROUTINE CALABC(IE,X,A)
C
C      if the element is found, where XI and ETA reside, then calculate
C      the coefficients for solving XI and ETA.
C
C      PARAMETER(NEMAX=600,NNODE=8)
C      COMMON /ELNOD/ LNODE(NEMAX,NNODE)
C
C      REAL*8 X(1),A(1)
C
C      I1=LNODE(IE,1)
C      I2=LNODE(IE,2)
C      I3=LNODE(IE,3)
C      I4=LNODE(IE,4)
C      A(1)= 0.25D0*(X(I1) + X(I2) + X(I3) + X(I4))
C      A(2)= -0.25D0*(X(I1) + X(I2) - X(I3) - X(I4))
C      A(3)= -0.25D0*(X(I1) - X(I2) - X(I3) + X(I4))
C      A(4)= 0.25D0*(X(I1) - X(I2) + X(I3) - X(I4))
C
C      RETURN
C      END
C-----
C      SUBROUTINE ROUND(X)
C
C      Due to numerical error, X may be slightly off what it is supposed
C      to be. If X is very close to 1, then let it be 1. If X is very
C      close to -1, then let it be -1.
C
C      REAL*8 X
C
C      IF (DABS(1.D0-DABS(X))<.1D-7) X=X/DABS(X)
C      RETURN
C      END
C-----
C      SUBROUTINE CALALF(XI,ETA,NNOD,IE,ALPHA)
C
C      after the XI and ETA had been found in LOCATE, calculate the
C      dy/dx at that point.
C
C      IMPLICIT REAL*8 (A-H,O-Z)

```

```

PARAMETER(NPMAX=1300,NEMAX=600,NNODE=8,LCCOD=2,IDER2=2,NDDOF=3)
COMMON /ELNOD/ LNODE(NEMAX,NNODE)
COMMON /COORD3/ XU(NPMAX),YU(NPMAX),ZU(NPMAX)
REAL*8 FUNC(4),DER2(2,4),DERIV(2,3),COORD(4,3)
CALL FMLIN2(2,DER2,IDER2,FUNC,XI,ETA)
CALL ECOORD(IE,COORD,NNOD,XU,YU,ZU)
CALL NULL(DERIV,LCCOD,LCCOD,NDDOF)
CALL MATMUL(DER2,COORD,DERIV,LCCOD,NNOD,NDDOF)
DET=DERIV(1,1)*DERIV(2,3)-DERIV(1,3)*DERIV(2,1)
IF (DET.LT.1.D-30) THEN
  WRITE (6,*) ' Jacobian determinant is',DET,' in CALALF'
  STOP
ENDIF
TANALF=(DERIV(2,3)*DERIV(1,2)-DERIV(1,3)*DERIV(2,2))/DET
ALPHA=DATAN(TANALF)
RETURN
END
C-----
C      SUBROUTINE CALXYZ(XI,ETA,NNOD,IE,X,COORD)
C
C      after the XI and ETA had been found in LOCATE, calculate the
C      other coordinate (see LOCATE for detail).
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      PARAMETER (NEMAX=600,NNODE=8,LCCOD=2)
C      COMMON /ELNOD/ LNODE(NEMAX,NNODE)
C
C      REAL*8 X(1)
C      REAL*8 FUNC(4),DER2(2,4)
C
C      CALL FMLIN2(1,DER2,LCCOD,FUNC,XI,ETA)
C      COORD=0.D0
C      DO 1 J=1,NNOD
C      K=LNODE(IE,J)
C      COORD=COORD+FUNC(J)*X(K)
C      1 CONTINUE
C      RETURN
C      END
C-----
C      SUBROUTINE TRANS(X1,Y1,X0,Y0,X2,Y2)
C
C      This subroutine translates all the coordinates for a
C      given slip-sinkage pivot or initial contact point.
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      X2=X1 - X0
C      Y2=Y1 - Y0
C      RETURN
C      END
C-----

```

```

C SUBROUTINE ROTATE(X1,Y1,OMEGA,X2,Y2)
C This subroutine rotates all the coordinates for a given
C slip-sinkage angle, Omega.
C
C IMPLICIT REAL*8 (A-H,O-Z)
C
C TMPS=DSIN(OMEGA)
C TMPC=DCOS(OMEGA)
C X2= X1*TMPC - Y1*TMPS
C Y2= X1*TMPS + Y1*TMPC
C RETURN
C END
C
C SUBROUTINE ECOOR(IE,COORD,NNOD,X,Y,Z)
C This subroutine assigns the input global coordinates to each
C element
C
C IMPLICIT REAL*8 (A-H,O-Z)
C PARAMETER(NEMAX=600,NNODE=8)
C COMMON /ELNOD/ LNODE(NEMAX,NNODE)
C
C REAL*8 X(*),Y(*),Z(*),COORD(NNOD,*)
C
C DO 10 I=1,NNOD
C K=LNODE(IE,I)
C COORD(1,I)=X(K)
C COORD(2,I)=Y(K)
C COORD(3,I)=Z(K)
C 10 CONTINUE
C RETURN
C END
C
C SUBROUTINE NULVEC(VEC,N)
C This subroutine initializes a vector [A] with zeroes
C
C REAL*8 VEC(*)
C
C DO 1 I=1,N
C VEC(I)=0D+0
C 1 CONTINUE
C RETURN
C END
C
C SUBROUTINE MULVEC(A,B,C,N)
C This subroutine multiplies the vector B by C
C and assigns it to vector A
C
C REAL*8 A(*),B(*),C

```



```

      ID=INC*(J,J)
      IF (YU(ID).LE.YSURF) GOTO 21
25  CONTINUE
      GOTO 20
21  K=K+1
      CNTSRF(K)=1
20  CONTINUE
      NECNT=K
      C --- reverse the order for later use
      DO 10 I=1,NECNT
        L=NECNT-I+1
        CNTELMI=L-CNTSRF(L)
10  CONTINUE
      RETURN
      END
      C-----
      SUBROUTINE CNVTIPS(COORD,NDDOF,NINOD,TRACT,SHEAR,FP,F,SH)
      C
      C This subroutine converts the normal pressure and shear stress
      C applied to an element to nodal forces in global coordinate system
      C
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (LCCOD=2,IDER2=2)
      COMMON /COEFFS/ COEFF(10,3),SAMP(2,2),NYCF(3),NGP
      C
      REAL*8 COORD(NINOD,3)
      REAL*8 FP(NINOD,3),FSH(NINOD,3),TRACT(3),SHEAR(3)
      REAL*8 DERIV(2,3),DER2(2,4),FUN2(4),A(3)
      C
      CALL NULL(FP,NINOD,NDDOF)
      CALL NULL(FSH,NINOD,NINOD,NDDOF)
      C
      DO 5 ID=1,NINOD
        DO 10 J=1,NGP
          DO 20 K=1,NGP
            XI=SAMP(J,1)
            ETA=SAMP(K,1)
            CALL FMLIN2(3,DER2,IDER2,FUN2,XI,ETA)
            CALL NULL(DERIV,LCCOD,LCCOD,NDDOF)
            CALL MATMUL(DER2,COORD,DERIV,LCCOD,NINOD,NDDOF)
            PRD=SAMP(J,2)*SAMP(K,2)
            A(1)=DERIV(1,2)*DERIV(2,3)-DERIV(1,3)*DERIV(2,2)
            A(2)=DERIV(1,3)*DERIV(2,1)-DERIV(2,1)*DERIV(2,3)
            A(3)=DERIV(1,1)*DERIV(2,2)-DERIV(1,2)*DERIV(2,1)
          C
          C --- convert normal pressure
          DG 50 IP=1,NDDOF
          FP(ID,IP)=FP(ID,IP)+FUN2(ID)*TRACT(ID)*A(IP)*PRD
          C
          C --- convert shear stress
          AM=DSORT(A(1)*A(1) + A(2)*A(2) + A(3)*A(3))
          C
          C
50  CONTINUE
      C
      C

```

```

      ETAP=1.D0+ETA
      IF (UO8.EQ.2) GOTO 10
      C --- Shape functions
      FUN2(1)= 0.25D0*XIM*ETAM
      FUN2(2)= 0.25D0*XIM*ETAP
      FUN2(3)= 0.25D0*XIP*ETAP
      FUN2(4)= 0.25D0*XIP*ETAM
      IF (UO8.EQ.1) RETURN
      C --- Derivatives of qza
      DO 10 DER2(1,1)=-0.25D0*ETAM
        DER2(1,2)=-0.25D0*ETAP
        DER2(1,3)= 0.25D0*ETAP
        DER2(1,4)= 0.25D0*ETAM
      C --- Derivatives of eia
      DO 10 DER2(2,1)=-0.25D0*XIM
        DER2(2,2)= 0.25D0*XIM
        DER2(2,3)= 0.25D0*XIP
        DER2(2,4)=-0.25D0*XIP
      C
      RETURN
      END
      C-----
      SUBROUTINE GAUSS
      C
      C This subroutine sets Gaussian integration parameters
      C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /COEFFS/ COEFF(10,3),SAMP(2,2),NYCF(3),NGP
      C
      SAMP(1,1)=1.D0/DSQRT(3.D0+0)
      SAMP(2,1)=SAMP(1,1)
      SAMP(1,2)=1.D0
      SAMP(2,2)=1.D0
      RETURN
      END
      C-----
      SUBROUTINE DETCNT(NE,CNTELM,NECNT,XU,YU,ZU)
      C
      C This subroutine determines contact elements
      C
      PARAMETER (NPMAX=1300,NEMAX=600,NNODE=8,NINOD=4)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ELNOD/ LNODE(NEMAX,NNODE)
      COMMON /SURFAC/ YSURF,YMIN,XBOT,RE THETA0,THETA1,DTHETA
      C
      REAL*8 XU(1),YU(1),ZU(1)
      INTEGER CNTELM(NEMAX),CNTSRF(NEMAX)
      C
      C --- detect the contact elements
      K=0
      DO 20 I=1,NE
        DO 25 J=1,NINOD

```

```
OMEGA=DATAN(A(1)/A(2))  
FSH(ID,1)=FSH(ID,1)+FUN2(ID)*SHEAR(ID)*AM*PRD*DCOS(OMEGA)  
FSH(ID,2)=FSH(ID,2)+FUN2(ID)*SHEAR(ID)*AM*PRD*JOIN(CA,CA)  
20 CONTINUE  
10 CONTINUE  
5 CONTINUE  
RETURN  
END
```

```

PROGRAM INTERP
C
C This program interpolates the output files from TTIM
C
C DIMENSION X0(4),YR(4,10),YW(20)
C CHARACTER*10 FLE1,FLE2,FLE3,FLE4,STRING*70
C
C four output files are selected for interpolation,
C normal reactions should be around X,
C NVP is the number of data points in each file
C
C OPEN(1,FILE='interp.in')
C READ(1,*) FLE1
C READ(1,*) FLE2
C READ(1,*) FLE3
C READ(1,*) FLE4
C READ(1,*) X
C READ(1,*) NNP
C CLOSE(1)
C
C OPEN(11,FILE=FLE1)
C OPEN(12,FILE=FLE2)
C OPEN(13,FILE=FLE3)
C OPEN(14,FILE=FLE4)
C
C read strings from four files and write that string
C to output file INTERP.OUT
C
C OPEN(20,FILE='interp.out')
C DO 1 I=1,14
C   READ(1,*) X0(I),YR(J,1),YR(J,2),YR(J,3)
C   1 CONTINUE
C   WRITE(20,*) X0(I),YR(J,1),YR(J,2),YR(J,3)
C
C read and interpolate X0 and YR, save in YW
C write them to INTERP.OUT
C
C DO 2 I=1,14
C   J=1,10
C   READ(1,*) X0(J),YR(J,1),YR(J,2),YR(J,3)
C   2 CONTINUE
C   DO 3 I=1,3
C     CALL LAGR13(X0,YR(1,I),4,X,YW(I))
C   3 CONTINUE
C   WRITE(20,*) X,YW(1),YW(2),YW(3)
C
C read strings from four files and write that string
C to output file INTERP.OUT
C
C DO 4 I=1,14
C   READ(1,*) X0(I),YR(J,1),YR(J,2),YR(J,3)
C   4 CONTINUE

```

```

WRITE(20,*) X0(I),YR(J,1),YR(J,2),YR(J,3)
C
C read and interpolate YR, save in YW
C write them to INTERP.OUT
C
C DO 5 I=1,14
C   J=1,10
C   READ(1,*) YR(J,1),YR(J,2),YR(J,3)
C   5 CONTINUE
C   DO 10 I=1,3
C     CALL LAGR13(X0,YR(1,I),4,X,YW(I))
C   10 CONTINUE
C   WRITE(20,*) YW(1),YW(2),YW(3)
C
C DO 6 I=1,14
C   READ(1,*) X0(I),YR(J,1),YR(J,2),YR(J,3)
C   6 CONTINUE
C   WRITE(20,*) X0(I),YR(J,1),YR(J,2),YR(J,3)
C
C read and interpolate YR, and save in YW
C write them to INTERP.OUT
C
C DO 7 I=1,NNP
C   DO 8 J=1,14
C     JI=J-10
C     READ(JI,*) JNUM,YR(JI,1),YR(JI,2),YR(JI,3)
C     8 CONTINUE
C     DO 9 JI=1,3
C       CALL LAGR13(X0,YR(1,JI),4,X,YW(JI))
C     9 CONTINUE
C     WRITE(20,200) JNUM,YW(1),YW(2),YW(3)
C     7 CONTINUE
C     200 FORMAT(5X,I4,3(F12.5,2X))
C     CLOSE(11)
C     CLOSE(12)
C     CLOSE(13)
C     CLOSE(14)
C
C CLOSE(20)
C STOP
C END
C
C-----
C SUBROUTINE LAGR13(X0,Y0,N,X,Y)
C
C Lagrange interpolation
C
C DIMENSION X0(N),Y0(N)
C
C I=1
C 10 IF (XLT 0.5*(X0(I+1)+X0(I+2))) GOTO 30
C IF (X.GE 0.5*(X0(N-2)+X0(N-1))) GOTO 20

```

```

I=I+1
GOTO 10
20 I=N-2
30 M=I+2
Y=0.0
DO 60 J=I,M
  P=1.0
  DO 50 K=I,M
    IF (J.NE.K) P=P*(X-X0(K))/(X0(J)-X0(K))
  50 CONTINUE
  Y=Y+P*Y0(J)
60 CONTINUE
RETURN
END
C

```

```

COMMON /OTHER/ PI,RATIO,SLIP,ANG,P12,NP,SCT
COMMON /SHRTP/ XBOT0,YBOT0,NSHTP,NSLPSK
COMMON /SKGPM/ AJAR,BIBR,PKI,PKR,EPSSKG
COMMON /PRFRM/ TOTFY,THRST,RESIS,ZSNK,KSRC
COMMON /COEFFS/ COEFF(10,2),SAMP(2,2),NYCF(2),NGP
REAL*8 LAMDA

C
PI=4 D0*DATAN(1 D0)
D2R=PI/180 D0
PERCT=.0 01D0
OPEN (9,FILE='gene3d.in')
READ (9, '(A40)') ITITLE
READ (9,') PRESS,FRCFAC
READ (9,') NP,NE,IP1

C
C read coordinates and boundary conditions
C
DO 10 I=1,NP
  READ(9,') IDUMY,XQ(I),YQ(I),ZQ(I)
10 CONTINUE
C --- convert cm to m
CALL MULVEC(XQ,XQ,PERCT,NP)
CALL MULVEC(YQ,YQ,PERCT,NP)
CALL MULVEC(ZQ,ZQ,PERCT,NP)

C
C read node numbers
C
DO 20 I=1,NE
  READ (9,') IDUMMY,(LNODE(I),K),K=1,NNODE)
20 CONTINUE
C
C read terrain properties in file TERRAIN.DAT
C
OPEN (19,FILE='terrain.dat')
C
C read pressure-sinkage data
C
READ (19,') KC,KPHI,NN,BB
BB=BB*PERCT
READ (19,') A0,AU
READ (19,') YSURF,DINCR,YLOAD
C
C read polynomials for Young's moduli, E_t and E_s
C
READ (19,') NYCF(1)
READ (19,') (COEFF(J,1),J=NYCF(1),1,-1)
READ (19,') NYCF(2)
READ (19,') (COEFF(J,2),J=NYCF(2),1,-1)
READ (19,')
READ (19,')

C
C read shear parameters
C

```

```

PROG:RAM.FNLSTR
C
C This program calculates the final pressure and shear stress distributions
C after running INTERP
C
C input files required INTERP.OUT, CENE3D.IN, TERRAIN.DAT
C
IMPLICIT REAL*8 (A-H,O-Z)
PARAMETER (NPMAX=1300,NEMAX=600,NNODE=8,NNOD=4,NDDOF=3)
COMMON /COORD3/ XU(NPMAX),YU(NPMAX),ZU(NPMAX)
COMMON /COORD1/ XQ(NPMAX),YQ(NPMAX),ZQ(NPMAX)
COMMON /ELNOD/ LNODE(NEMAX,NNODE)
COMMON /MEINF/ NP,NE,NM,NB

C
REAL*8 KC,KPHI,NN
COMMON /TERRAIN/ KC,KPHI,NN,BB,A0,AU
REAL*8 KINT,KRUB
COMMON /SHRINF/ CINT,TANFII,KINT,CRUB,TANFIR,KRUB
COMMON /SURFAC/ YSURF,YMIN,XBOT,RBAR,THETA0,THETA1,DTHETA
COMMON /OTHER/ PI,RATIO,SLIP,ANG,P12,NP,SCT
COMMON /SHRTP/ XBOT0,YBOT0,NSHTP,NSLPSK
COMMON /PRFRM/ TOTFY,THRST,RESIS,ZSNK,KSRC
COMMON /COEFFS/ COEFF(10,2),SAMP(2,2),NYCF(2),NGP
CHARACTER*40 ITITLE

C
NGP=2
CALL GAUSS
C
CALL DATIN(ITITLE,PRESS,FRCFAC,IP1)
C
CALL SLPSNK(ITITLE,PRESS,FRCFAC,IP1)
C
STOP
END
C-----
SUBROUTINE DATIN(ITITLE,PRESS,FRCFAC,IP1)
C
C This subroutine reads input data
C
IMPLICIT REAL*8 (A-H,O-Z)
CHARACTER*40 ITITLE
PARAMETER (NPMAX=1300,NEMAX=600,NNODE=8,NDDOF=3,NMMAX=2)
COMMON /COORD3/ XU(NPMAX),YU(NPMAX),ZU(NPMAX)
COMMON /COORD1/ XQ(NPMAX),YQ(NPMAX),ZQ(NPMAX)
COMMON /ELNOD/ LNODE(NEMAX,NNODE)
COMMON /MEINF/ NP,NE,NM,NB

C
REAL*8 KC,KPHI,NN
COMMON /TERRAIN/ KC,KPHI,NN,BB,A0,AU
REAL*8 KINT,KRUB
COMMON /SHRINF/ CINT,TANFII,KINT,CRUB,TANFIR,KRUB
COMMON /SURFAC/ YSURF,YMIN,XBOT,RBAR,THETA0,THETA1,DTHETA

```



```

THRST=THRST*2.D0
RESIS=RESIS*2.D0

C
C      -- output the final results
C
OPEN (18,FILE='final.out')
WRITE (18,*) 'Normal reaction   thrust   resistance'
WRITE (18,*) TOTFY, THRST, RESIS
WRITE (18,*) 'Sinkage   YSURF   slip-sinkage   Omega'
WRITE (18,*) (YSURF-YMIN), YSURF, ZSNK, OMEGA
WRITE (18,*) 'New Coordinates for final calculation'

C
C      write deformed body coordinates
C
DO 4 I=1,NP
  IF (NSLPSK.EQ.0) THEN
    WRITE (18,50) I,XU(I)*100.D0,YU(I)*100.D0,ZU(I)*100.D0
  ELSE
    WRITE (18,50) I,XR(I)*100.D0,YR(I)*100.D0,ZU(I)*100.D0
  ENDIF
4 CONTINUE

C
DO 20 I=1,NE
  WRITE (18,25) I,(LNODE(I,K),K=1,NNODE)
20 CONTINUE
25 FORMAT(9(5))

C
C      write pressure and shear stress distributions
C
WRITE (18,*) ' Pressure and Shear Stress Distributions'
50 FORMAT(5X,I4,3(F12.5,2X))

C
DO 70 I=1,NECNT
  IF (NSLPSK.EQ.1) IE=CNTELM(I)
  IF (NSLPSK.EQ.0) IE=CNTRSF(I)
  CALL SETMP(IE,IP,IPS,SAVP,SAVSHR,TRACT,SHEAR,IL)

C
C      ... if no contact point in this element, skip
C
  IF (IL.EQ.0) GOTO 70

  IF (NSLPSK.EQ.1) CALL ECOORD(IE,COORD,NNOD,XR,YR,ZU)
  IF (NSLPSK.EQ.0) CALL ECOORD(IE,COORD,NNOD,XU,YU,ZU)
  WRITE (18,80) IE
  WRITE (18,81) (LNODE(IE,J),J=1,NNOD)
  DO 60 K=1,NNOD
    PRE=TRACT(K)
    WRITE (18,82) (COORD(K,J)*100.D0,J=1,NDOF),PRE,SHEAR(K)
  60 CONTINUE
  70 CONTINUE
  CLOSE(18)

80 FORMAT(1X,Element: ',3)
81 FORMAT(1X,4(14,2X))

82 FORMAT(1X,3(F12.5,2X),2(G12.5,2X))

C
OPEN(28,FILE='gene3dr.in')
WRITE(28,'(A)') ITITLE
WRITE(28,'(2(F10.5))') PRESS,FRCFAC
WRITE(28,'(3(5))') NP,NE,IP1

C
C      write undeformed but transformed coordinates for comparison
C
DO 100 I=1,NP
  WRITE(28,1000) I,XQ(I)*100.D0,YQ(I)*100.D0,ZQ(I)*100.D0
100 CONTINUE
DO 200 I=1,NE
  WRITE(28,2000) I,(LNODE(I,J),J=1,NNODE)
200 CONTINUE
1000 FORMAT(5,3(F10.5))
2000 FORMAT(9(5))
CLOSE(28)
RETURN
END

C-----
SUBROUTINE SETMP(IE,NIP,IPS,SAVP,SAVSHR,TRACT,SHEAR,IL)
C
C      This subroutine initializes
C      temporary pressure and shear stress arrays
C
C
PARAMETER (NPMAX=1300, NEMAX=600, NNODE=8, NNOD=4)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /ELNOD/ LNODE(NEMAX,NNODE)
REAL*8 SAVP(*),SAVSHR(*),TRACT(*),SHEAR(*)
INTEGER IPS(*)

C
  IL=0
  CALL NULVEC(TRACT,NNOD)
  CALL NULVEC(SHEAR,NNOD)
  DO 50 J=1,NNOD
    ID=LNODE(IE,J)
    DO 55 K=1,NIP
      IF (ID.EQ.IDPS(K)) THEN
        TRACT(J)=SAVP(K)
        SHEAR(J)=SAVSHR(K)
      GOTO 52
    ENDIF
  55 CONTINUE
  GOTO 50
  52 IL=IL+1
  50 CONTINUE
  RETURN
  END

C-----
SUBROUTINE FNDBOT(NP,X,Y,Z,XBOT,YBOT,ZBOT)
C

```


C This subroutine finds the lowest point of the system

```
C  
  REAL*8 X(*),Y(*),Z(*),YBOT,XBOT,ZBOT  
  YBOT=-1.D+30  
  DO 5 I=1,NP  
    IF (Y(I) LT YBOT) THEN  
      YBOT=Y(I)  
      XBOT=X(I)  
      ZBOT=Z(I)  
    ENDIF  
  5 CONTINUE  
  RETURN  
  END
```

Appendix E

Computer Graphics Source Codes for Finite Element Models

Program TRAC3P is designed to display finite element meshes in three-dimension on Sun Sparc II workstation.

Program SSP is designed to display normal pressure and shear stress distributions in three-dimension on Sun Sparc II workstation.

Program SV is designed to display normal pressure and shear stress distributions for given view sections in two-dimension on Sun Sparc II workstation.

```

C .....
C          TRACP3
C          -- TRAC Data Plot Program with 3D Graphic Routines --
C          Used Library : GKS
C          by Hiroshi Nakashima and Songlao Chen
C          Department of Mechanical and Aerospace Engineering
C          Carleton University
C          (Ver. 1.3, October, 1992)
C .....
C          To understand this software, knowledge of 3D computer graphics
C          and GKS libraries are essential.
C .....
C          INCLUDE /usr/local/lib/gks/gksdd.for
C          CHARACTER*80 ITITLE,IDFILE
C          CHARACTER*80 ILOGO,ILOGO1
C          CHARACTER*80 IDEVS(2)
C          CHARACTER*80 JNP
C          CHARACTER*5 IPROC,IREDO,H,CHNG
C          PARAMETER(NPIMAX=2000,NEMAX=1000,NSMAX=9000)
C          COMMON /COORD/ X(NPIMAX),Y(NPIMAX),Z(NPIMAX)
C          COMMON /COORD1/ SX(NPIMAX),SY(NPIMAX),SZ(NPIMAX)
C          COMMON /ELNOD/ L(NODE(NEMAX,8))
C          COMMON /MEINF/ NP,NE
C          COMMON /VIEW/ X1,Y1,Z1,S1,C1,S2,C2
C          COMMON /SURF/ ISUR,ISURF(NSMAX,5),JSUR,JSURF(NSMAX,5)
C          COMMON /WORKS/ ICONT(NEMAX,6,5),JCONT(NEMAX,6)
C          COMMON /NORMLS/ VNORM(NSMAX,3)
C          COMMON /EDGES/ IED,IEDGE(NSMAX,2)
C          COMMON /PARAM/ D2R,R2D
C          COMMON /VISIV/ IVISINPIMAX)
C          COMMON /PLOTS/ IPL,XE(NPIMAX,2),YE(NPIMAX,2)
C          REAL          LWIDTH
C          INTEGER       PLI,COLI
C          COMMON /GRAPH/ ORIGX,ORIGY,AXLEN,TXLEN,CHAHT,
C          &          IFONT,I,TYPE,PLI,COLI,LWIDTH
C .....
C          DIMENSION CMESH(2,2)
C          DIMENSION TEXTPT(2)
C          INTEGER ASFLST(4)
C          there are two output devices, screen and postscript file
C .....
C          IDEVS(1)='screen'
C          IDEVS(2)='gks2ps'
C          the following text will appear on all the diagrams
C          produced by this program
C .....
C          ILOGO='Transport Technology Research Laboratory'
C          ILOGO1='Carleton University, Ottawa, CANADA'

```

```

PI=4.0*ATAN(1.)
D2R=PI/180.
R2D=180.0/PI

C
C      Data Input Menu
C
C      gene3d : FEM mesh
C      contact : deformed FEM mesh
C
C      WRITE (':') '// Welcome to T R A C P 3 D //'
C      WRITE (':')
C      WRITE (':') /Data File Selection/
C      WRITE (':') '1:gene3d 2:contact'
C      WRITE (':') '[Your data] ...'
C      READ (':') IFILE
C
C      IF (IFILE.EQ.1) THEN
C      CALL DATIN(ITITLE)
C      ELSE IF (IFILE.EQ.2) THEN
C      WRITE (':') ' > Enter Stage Number...'
C      READ (':') ISTAGE
C      IF (INT(ISTAGE/10).GE.1) THEN
C      WRITE(IDFILE,96) ISTAGE
C      ELSE
C      WRITE(IDFILE,97) ISTAGE
C      WRITE(IDFILE,97) ISTAGE
C      FORMAT('contact',I1,'.disp')
C      ENDIF
C      CALL DATIN(ITITLE)
C      CALL DATA3(IDFILE,ITITLE,ZMAX)
C      ENDF
C
C      WRITE(JNP,1200) NP,NE
C      1200 FORMAT('Total Nodes:',I5,5X,'Total Elements:',I5)
C
C      call subroutine SURFAC to determine visible surfaces
C      (some sides of an element, non-common surfaces)
C
C      CALL SURFAC
C
C      Set Window size (default is 200x200).
C      origin is set at the center of the window.
C      Set Viewport size as default.
C
C      WYMIN=-100.0
C      WYMAX= 100.0
C      WYMIN=-100.0
C      WYMAX= 100.0
C      VXMIN= 0.0
C      VYMAX= 1.0
C      VYMIN= 0.0

```

```

VYMAX= 1.0
C
C Menu of Job
C
C 1 : will display the diagram on screen
C
C 2 : will write the diagram into a postscript file
C named GKS2PS.OUTPUT
C
C 3 : will quit the program
C
C
C IWIN1=0
C IWIN2=0
C CHNG='Y'
C 400 WRITE (':')
C WRITE (':') '3D Graphics Menu'
C WRITE (':') '1:CRT 2:PS File 3:Exit'
C WRITE (':') '[Your job]...'
C READ (':') LOB
C IF ((LOB.EQ.1).OR. (LOB.EQ.2)) GOTO 500
C IF ((LOB.EQ.3).OR. (LOB.EQ.0)) GOTO 700
C GOTO 400
C
C CRT Display and Postscript File Output
C
C 500 CONTINUE
C
C Open GKS Routines for LOB=2 after LOB=1
C
C IF ((IWIN1.EQ.1) OR. (IWIN2.EQ.1)) THEN
C 250 WRITE (':') 'Anything to change ? (Y/N)'
C READ (':(A)') CHNG
C IF (CHNG.EQ.'Y') CHNG='Y'
C IF (CHNG.EQ.'N') CHNG='N'
C IF ((CHNG.NE.'Y') AND. (CHNG.NE.'N')) GOTO 250
C IF ((LOB.EQ.2). AND. (IWIN1.EQ.1)) WRITE (':',800)
C CALL GDAWK(1)
C CALL GCLWK(1)
C CALL GCLKS
C ENDIF
C
C IF (LOB.EQ.1) THEN
C WRITE (':') '[CRT Display]'
C IWIN2=0
C IWIN1=1
C ELSE
C WRITE (':') '[PS File Creation]'
C IWIN1=0
C IWIN2=1
C ENDIF
C
C Open GKS Routines for General Case
C
C CALL OPNGKS(2)

```

```

DO 1 I=1,4
ASFLST(I)=GINDIV
1 CONTINUE
CALL SETASF(ASFLST)
CALL OPNWSN(1.0,IDEVS(LOB))
CALL GACWK(1)
CALL GSVPR(1,VXMIN,VXMAX,VYMIN,VYMAX)
CALL GSWN(1,WXMIN,WXMAX,WYMIN,WYMAX)
CALL GSELNT(1)
CALL GSPLR(1,PL1,TYPE,LWIDTH,COL)
CALL GSTXFP(IFONT,1)
C
C IF (CHNG.EQ.'N') GOTO 350
C
C select one of the following options for viewing
C
C Eye point : location of eye in the same coordinate
C system where the object is displayed
C Rotation angle: absolute angle with horizontal, rotate
C the picture not the object
C Scale factor : scale the picture not the object
C Hidden : ask whether hide those lines that are
C not supposed to be seen.
C The status will remain until next
C selection is made The default is No
C
C WRITE (':') '> Eye Point: x,y,z'
C READ (':') X1,Y1,Z1
C IF (CHNG.EQ.'Y') AND. (LOB.EQ.2)) THEN
C WRITE (':') '> Rotation angle'
C READ (':') THETA
C STHETA=SIN(THETA*D2R)
C CTHETA=COS(THETA*D2R)
C WRITE (':') '> Scale factor'
C READ (':') FACTR
C WRITE (':') '> Hidden ? (Y/N)'
C READ (':(A)') IH
C IF (IH.EQ.'Y') IH='Y'
C IF (IH.NE.'Y') IH='N'
C ELSE
C STHETA=0.0
C CTHETA=1.0
C FACTR=1.0
C IH='N'
C ENDIF
C
C Determine the visible surfaces and their
C normal vectors.
C
C 300 CONTINUE
C CALL NORMAL
C

```

```

C      determine visible edges by calling
C      subrou. to CEDGE.
C
C 360 CONTINUE
C      CALL CEDGE(IH)
C
C      Hidden Line Processing and Clipping
C
C      IF (H1.EQ. 'Y') THEN
C          DO 37 I=1,NP
C              IVIS(I)=0
C          37 CALL HIDDEN
C      ELSE
C          IPL=IED
C      ENDIF
C
C 350 IF (J08.EQ.2) THEN
C      WRITE ('.') ' (PS File is prepared)'
C      ENDIF
C
C      CALL GCLRWK(1,GALWAY)
C
C      DO 99 I=1,IPL
C          --- assign lines
C          IF (IH.EQ. 'Y') THEN
C              CMESH(1,1)=XE(I,1)
C              CMESH(2,1)=YE(I,1)
C              CMESH(1,2)=XE(I,2)
C              CMESH(2,2)=YE(I,2)
C          ELSE
C              K=IEDGE(I,1)
C              L=IEDGE(I,2)
C              CMESH(1,1)=SX(K)
C              CMESH(2,1)=SY(K)
C              CMESH(1,2)=SX(L)
C              CMESH(2,2)=SY(L)
C          ENDIF
C
C          --- rotate lines
C
C      CALL ROTAT(CMESH,STHETA,CTHETA)
C
C          --- scale lines
C
C      CMESH(1,1)=CMESH(1,1)*FACR
C      CMESH(2,1)=CMESH(2,1)*FACR
C      CMESH(1,2)=CMESH(1,2)*FACR
C      CMESH(2,2)=CMESH(2,2)*FACR
C
C      CALL GPL(2,CMESH)
C      99 CONTINUE
C
C      --- write some texts
C      CALL GSTXAL(2,0)
C      TEXTPT(1)=0
C      TEXTPT(2)=81.
C      CALL GSCHH(7.)
C      CALL TEXT(TEXTPT,ITITLE)
C      TEXTPT(2)=87.
C      CALL GSCHH(6.)
C      CALL TEXT(TEXTPT,JNP)
C      TEXTPT(2)=92.
C      CALL GSCHH(5.)
C      CALL TEXT(TEXTPT,ILOGO)
C      TEXTPT(2)=97
C      CALL TEXT(TEXTPT,ILOGO1)
C
C      --- draw the axes
C      CALL AXES(STHETA,CTHETA)
C
C      --- update workstation
C      CALL GUNWK(1,GSUPPD)
C
C      --- modify if necessary
C      IF (J08.EQ.1) THEN
C          450 WRITE ('.') ' >> [R]edo, or [Q]uit'
C          READ ('(A5)') IPROC
C          IF ((IPROC.EQ.'Q') .OR. (IPROC.EQ.'q')) GOTO 400
C          IF ((IPROC.EQ.'R') .OR. (IPROC.EQ.'r')) THEN
C              select one of the following options for viewing
C
C              Eye point : location of eye in the same coordinate
C                          system where the object is displayed
C              Rotation angle: absolute angle with horizontal, rotate
C                          the picture not the object
C              Scale factor : scale the picture not the object
C              Hidden : ask whether hide those lines that are
C                      not supposed to be seen.
C                      The status will remain until next
C                      selection is made. The default is No
C
C          450 WRITE ('.') ' >> [E]ye point'
C              WRITE ('.') ' [R]otation'
C              WRITE ('.') ' [S]cale'
C              WRITE ('.') ' [H]idden'
C              READ ('(A5)') IREDO
C              IF ((IREDO.EQ.'E') .OR. (IREDO.EQ.'e')) THEN
C                  WRITE ('.') ' > Eye Point: x,y,z'
C                  READ ('.') X1,Y1,Z1
C                  GOTO 300
C              ELSEIF ((IREDO.EQ.'R') .OR. (IREDO.EQ.'r')) THEN
C                  WRITE ('.') ' > Rotation angle'
C                  READ ('.') THETA

```

```

      STHETA=SIN(THETA'D2R)
      CTHETA=COS(THETA'D2R)
      GOTO 350
    ELSEIF ((IREDO.EQ.'S') .OR. (IREDO.EQ.'s')) THEN
      WRITE ('.') ' Scale factor'
      READ ('.') FACTR
      GOTO 350
    ELSEIF ((IREDO.EQ.'H') .OR. (IREDO.EQ.'h')) THEN
      WRITE ('.') ' Yes or No ?'
      READ ('.') IH
      IF (IH.EQ.'Y') IH='Y'
      IF (IH.EQ.'N') IH='N'
      IF ((IH.EQ.'Y') .AND. (IH.NE.'N')) GOTO 480
      GOTO 360
    ELSE
      GOTO 460
    ENDIF
  ENDIF
  GOTO 450
ENDIF
GOTO 400
C
700 CONTINUE
C
C      ... finish up
C
      IF ((WIN1.EQ.1) WRITE(' ',800)
800 FORMAT('X: ' > Press Any Key within GKS Window ')
      CALL GDAWK(1)
      CALL GCLWK(1)
      CALL GCLKS
      STOP
      END
C
      SUBROUTINE DATAIN(ITITLE)
C
C      This subroutine reads input data stored in file
C      GENE3D.DAT
C
      PARAMETER(NPMAX=2000,NEMAX=1000,NNODE=8)
      COMMON /COORD/ X(NPMAX),Y(NPMAX),Z(NPMAX)
      COMMON /ELNOD/ LNODE(NEMAX,8)
      COMMON /AEINF/ NP,NE
      REAL          LWIDTH
      INTEGER       PLI,COLI
      COMMON /GRAPH/ ORIGX,ORIGY,AXLEN,TXLEN,CHAHT,
&      FONT,LTYPE,PLI,COLI,LWIDTH
      CHARACTER*80 ITITLE
C
      WRITE ('.') ' DATAIN'
      OPEN (9,FILE='gene3d.dat')
      READ (9,('A40')) ITITLE
      READ (9,') PRESS:FRCFAC

```

```

      READ (9,') NP,NE,NDUMMY
C
C      find the limits of the coordinates
C
      DMAX=0.
      XMIN= 1.E+20
      XMAX=-1.E+20
      YMIN= 1.E+20
      YMAX=-1.E+20
      ZMIN= 1.E+20
      ZMAX=-1.E+20
      DO 200 I=1,NP
        READ (9,') IDUMY,X(I),Y(I),Z(I)
        IF (X(I).LT.XMIN) XMIN=X(I)
        IF (X(I).GT.XMAX) XMAX=X(I)
        IF (Y(I).LT.YMIN) YMIN=Y(I)
        IF (Y(I).GT.YMAX) YMAX=Y(I)
        IF (Z(I).LT.ZMIN) ZMIN=Z(I)
        IF (Z(I).GT.ZMAX) ZMAX=Z(I)
        TEMP=AMAX1(ABS(X(I)),ABS(Y(I)),ABS(Z(I)))
        IF (TEMP.GT.DMAX) DMAX=TEMP
      200 CONTINUE
C
C      read FEM mesh numbering information
C
      DO 220 I=1,NE
        READ (9,') IDUMY,(LNODE(I,K),K=1,NNODE)
      220 CONTINUE
      CLOSE(9)
C
C      read some parameters for plotting
C      WIND      window length in user unit
C      PLI      polyline index
C      LTYPE     polyline type
C      LWIDTH    polyline width scale factor
C      COLI      color index
C      IFONT     text font
C      ORIGX, ORIGY . x, y coordinates of the axes in user unit
C      AXLEN     axis length
C      TXLEM     distance of text X, Y and Z from the origin
C      TXTLEN    text height for X, Y and Z in user unit
C      CHAHT     character height
C
      OPEN (19,FILE='trac3gl.dat')
      READ (19,') WIND
      READ (19,') PLI,LTYPE
      READ (19,') LWIDTH,COLI
      READ (19,') IFONT
      READ (19,') ORIGX,ORIGY
      READ (19,') AXLEN,TXLEN
      READ (19,') CHAHT
      CLOSE (19)

```

```

C      --- define new origin at center of the body,
C      translate the coordinates
C
      IF (NORIG.EQ.0) THEN
        XORIG=0.5*(XMIN+XMAX)
        YORIG=0.5*(YMIN+YMAX)
        ZORIG=0.5*(ZMIN+ZMAX)
      ENDIF
      DO 100 I=1,NP
        X(I)=X(I)-XORIG
        Y(I)=Y(I)-YORIG
        Z(I)=Z(I)-ZORIG
      100 CONTINUE
C
C      scale user coordinates to default coordinates.
C      default window is 200x200, its origin is located
C      at the center of the window.
C      leave 20% space as margin.
C
      SCALE=.80/0.0*WIND
      DO 300 I=1,NP
        X(I)=X(I)*SCALE
        Y(I)=Y(I)*SCALE
        Z(I)=Z(I)*SCALE
      300 CONTINUE
C
C      scale the parameters for plotting the axes
C
      SCALE=100.0/WIND
      ORIGX=ORIGX*SCALE
      ORIGY=ORIGY*SCALE
      AXLEN=AXLEN*SCALE
      TXLEN=TXLEN*SCALE
      CHAHT=CHAHT*SCALE
C
      RETURN
      END
C-----
C      SUBROUTINE DATA3(IDFILE,ITITLE,ZMAX)
C
C      This subroutine reads the deformed FEM mesh
C      (coordinate translation and scaline have not been done)
C
      CHARACTER*80 ITITLE,IDFILE
      PARAMETER(NPMAX=2000,NEMAX=1000,NM,NMAX=4)
      COMMON /COORDX(NPMAX),Y(NPMAX),Z(NPMAX)
      COMMON /ELNOD1,NODE(NEMAX,8)
      COMMON /MEINF,NP,NE
      WRITE(*,'...DATAIN[Contact]')
      OPEN(9,FILE=IDFILE)
      READ(9,*(A80)) ITITLE
      READ(9,*) NP,NE

```

```

ZMAX=0.
DO 200 I=1,NP
  READ(9,*) L,YY,ZZ,XX,UY,UZ,UX
  X(I)=XX+JUX
  Y(I)=YY+JUY
  Z(I)=ZZ+JUZ
  IF(Z(I).GT.ZMAX) ZMAX=Z(I)
200 CONTINUE
CLOSE(9)
RETURN
END
C-----
C      SUBROUTINE SURFAC
C
C      This subroutine determines the visible surfaces, ie, those
C      of an element are not shared by any other elements.
C
      PARAMETER(NPMAX=2000,NEMAX=1000,NSMAX=9000)
      COMMON /ELNOD/ LNODE(NEMAX,8)
      COMMON /MEINF/ NP,NE
      COMMON /SURFS/ ISUR,ISURF(NSMAX,5),JSUR,JSURF(NSMAX,5)
      COMMON /WORKS/ ICONT(NEMAX,6,5),JCONT(NEMAX,6)
C
      WRITE(*,'...SURFAC')
      DO 40 I=1,NE
C
C      put the node number of all surfaces of an element
C      in array ICONT in an order of right-hand rule
C      (see Fig 2)
C
        DO 45 K=1,5
          1-4-3-2-1
          J=K
          IF (K.EQ.2) J=4
          IF (K.EQ.4) J=2
          IF (K.EQ.5) J=1
          ICONT(I,1,K)=LNODE(I,J)
          5-6-7-8-5
          L=K+4
          IF (K.EQ.5) L=5
          ICONT(I,2,K)=LNODE(I,L)
          1-2-6-5-1
          L1=K
          IF (K.EQ.3) L1=6
          IF (K.EQ.4) L1=5
          IF (K.EQ.5) L1=1

```

```

CONTINUE
C
C      if all five points are equal for two
C      different elements, then it is safe to
C      say that this surface is shared by both
C      elements
C
      IF (L.EQ.5) THEN
        JCONT(I,J1)=1
        JCONT(J,J2)=1
        GOTO 20
      ENDIF
25  CONTINUE
20  CONTINUE
10  CONTINUE
C
C      save those non-shared surfaces in ISURF
C
      L=0
      DO 100 I=1,NE
        DO 200 K=1,6
          IF (JCONT(I,K).EQ.1) GOTO 200
          L=L+1
          DO 210 K1=1,5
            ISURF(L,K1)=JCONT(I,K,K1)
210    CONTINUE
200    CONTINUE
100    CONTINUE
      ISUR=L
      RETURN
      END
C-----
SUBROUTINE NORMAL
C
C      This subroutine projects the coordinates onto
C      normal plane and calculate the normal vectors of
C      all visible surfaces
C
      PARAMETER(NPMAX=2000,NEMAX=1000,NSMAX=9000)
      COMMON /COORD/ X(NPMAX),Y(NPMAX),Z(NPMAX)
      COMMON /VIEW/ X1,Y1,Z1,S1,C1,S2,C2
      COMMON /MEINF/ NP,NE
      COMMON /COORD1/ SX(NPMAX),SY(NPMAX),SZ(NPMAX)
      COMMON /SURFS/ ISUR(SURF(NSMAX,5)),JSUR(JSUR(NSMAX,5))
      COMMON /NORMLS/ VNORM(NSMAX,3)
      COMMON /PARAM/ D2R,R2D
C
      WRITE (*,' ') ' NORMAL '
C
C      convert eye point (x,y,z) in Cartesian coordinate system
C      to spherical coordinate system
C
      ICONT(I,3,K)=LNODE(IL,1)
C
C      2-3-7-6-2
C
      ICONT(I,4,K)=LNODE(IL,1+1)
C
C      4-8-7-3-4
C
      L5=K+3
      IF(K.EQ.2) L5=8
      IF(K.EQ.3) L5=7
      IF(K.EQ.4) L5=3
      IF(K.EQ.5) L5=4
      ICONT(I,5,K)=LNODE(IL,5)
C
C      1-5-8-4-1
C
      L2=K
      IF(K.EQ.2) L2=5
      IF(K.EQ.3) L2=8
      IF(K.EQ.5) L2=1
      ICONT(I,6,K)=L*JODE(IL,2)
45  CONTINUE
40  CONTINUE
C
C      initialize JCONT
C
      DO 5 I=1,NE
        DO 5 J=1,6
          JCONT(I,J)=0
5  CONTINUE
C
C      find out if surfaces are shared by two elements
C
      DO 10 I=1,NE-1
        DO 20 J=I+1,NE
          one element has six surfaces
C
          DO 25 J1=1,6
            DO 25 J2=1,6
              L=0
              DO 30 K=1,5
                L1 and L2 are nodes from two
                different elements.
                if they are equal, increase the counter L
C
                K2=6-K
                L1=JCONT(I,J1,K)
                L2=JCONT(J,J2,K2)
                IF (L1.EQ.L2) L=L+1

```



```

83  CONTINUE
C
C      from the visible surface, store the non-repeated
C      visible edges or line segments
C
      IE1=ISURF(I,1)
      DO 82 J=2,5
      IE2=ISURF(I,J)
      DO 84 K=1,N+1
C
C          if this line segment is the duplicate of
C          of a previously recorded one, then skip this one
C
          IF ((IEGE(K,1)EQ IE2) .AND.
&           (IEGE(K,2)EQ IE1)) GOTO 85
      84  CONTINUE
          N=N+1
          IEGE(N,1)=IE1
          IEGE(N,2)=IE2
      85  IE1=IE2
      82  CONTINUE
C
      80 CONTINUE
C
      IED=N
      RETURN
      END
C

```

A resolution test chart featuring various patterns of horizontal and vertical lines of different thicknesses. Each pattern is accompanied by a numerical value indicating the resolution level. The values include 1.0, 1.1, 1.25, 1.4, 1.6, 1.8, 2.0, 2.2, 2.5, 2.8, 3.2, 3.6, 4.0, 4.5, 5.0, 5.6, 6.3, 7.1, 8.0, 9.0, 10, 11.2, 12.5, 14, 16, 18, 20, 22.5, 25, 28, 32, 36, 40, 45, 50, 56, 63, 71, 80, 90, 100, 112, 125, 140, 160, 180, 200, 225, 250, 280, 320, 360, 400, 450, 500, 560, 630, 710, 800, 900, 1000, 1120, 1250, 1400, 1600, 1800, 2000, 2250, 2500, 2800, 3200, 3600, 4000, 4500, 5000, 5600, 6300, 7100, 8000, 9000, 10000.

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
(ANSI and ISO TEST CHART No. 2)

```

C.....
C      Spatial Surface Plot (SSP)
C      -- TTM stress distributions plotting program --
C      Used Library : GKS
C      by Songtao Chen
C      Department of Mechanical and Aerospace Engineering
C      Carleton University
C      (Ver. 1.0, December 10, 1991)
C.....
C      To understand this software, knowledge of 3D computer graphics
C      and GKS libraries are essential.
C.....
C      INCLUDE 'usr/local/lib/gks/gksdd.f'
C      PARAMETER(NPMAX=1000,NEMAX=500,NSMAX=1000,NNODE=4)
C      COMMON /COORD/ X(NPMAX),Y(NPMAX),Z(NPMAX)
C      COMMON /COORD1/ SX(NPMAX),SY(NPMAX),SZ(NPMAX)
C      COMMON /EINF/ NP,NE
C      COMMON /VIEW/ X1,Y1,Z1,S1,C1,S2,C2
C      COMMON /SURFS/ ISUR,ISURF(NSMAX,5)
C      COMMON /EDGES/ IED,IEDGE(NSMAX,2)
C      COMMON /PARAM/ D2R,R2D
C      COMMON /VISW/ VSI(NPMAX)
C      COMMON /PLOTS/ IPL,XE(NPMAX,2),YE(NPMAX,2)
C      REAL LWIDTH
C      INTEGER PL,COLI
C      COMMON /GRAPH/ ORIGX,ORIGY,AXLEN,TXLEN,CHAHT,
C      & FONT,LTYPE,PLI,COLI,LWIDTH
C      COMMON /DSCR/ XMIN,XMAX,YMIN,YMAX,SMAX
C
C      DIMENSION CMESH(2,2),TEXTPT(2)
C      CHARACTER*80 ILOGO,ILOGO1,ZCH
C      CHARACTER*80 IDEVS(2)
C      CHARACTER*80 JNP,ITITLE,XRANG,YRANG,PKVAL
C      CHARACTER*5 IPROC,IRED,CHNG,IH
C      INTEGER ASFLST(4)
C
C      there are two output devices, screen and postscript file
C
C      IDEVS(1)='screen'
C      IDEVS(2)='gs2ps'
C
C      the following text will appear on all the diagrams
C      produced by this program
C
C      ILOGO='Transport Technology Research Laboratory'
C      ILOGO1='Carleton University, Ottawa, CANADA'
C      PL=4 0*ATAN(1)
C      D2R=PI*180
C      R2D=180 0.PI
C

```

```

C      print headline and read input data.
C
C      WRITE(' ') // Welcome to Spatial Surface Plot (SSP) //
C      CALL DATAIN(ITITLE,ZCH)
C
C      Calculation of Edge Information
C
C      CALL CEDGE
C
C      Set Window size (default is 200x200),
C      origin is set at the center of the window.
C      Set Viewport size as default.
C
C      WXMIN=-100 0
C      WXMAX= 100 0
C      WYMIN=-100 0
C      WYMAX= 100 0
C      VXMIN= 0 0
C      VXMAX= 1 0
C      VYMIN= 0 0
C      VYMAX= 1 0
C
C      NE05=NE/2
C      WRITE(JNP,1200) NP,NFC5
C      FORMAT('Total Nodes ',I4,5X,'Total Facets ',I4)
C      WRITE(XRANG,1210) YMIN,XMAX
C      1210 FORMAT('X-range ',F9.2,' to ',F9.2,' cm')
C      WRITE(YRANG,1220) YMIN,YMAX
C      1220 FORMAT('Y-range ',F9.2,' to ',F9.2,' cm')
C      WRITE(PKVAL,1230) SMAX
C      1230 FORMAT('Peak value ',F9.2,' kPa')
C
C      Menu of Job
C
C      IWIN1=0
C      IWIN2=0
C      CHNG='Y'
C      400 WRITE (' ')
C      WRITE (' ') //3D Graphics Menu//
C      WRITE (' ') // 1 CRT 2 PS File 3 Exit//
C      READ (' ') //Your job//
C      READ (' ') //JOB//
C      IF ((JOB EQ 1) OR (JOB EQ 2)) GOTO 500
C      IF ((JOB EQ 3) OR (JOB EQ 0)) GOTO 700
C      GOTO 400
C
C      CRT Display and Postscript File Output
C
C      500 CONTINUE
C
C      Open GKS Routines for JOB=2 (postscript file)
C      after JOB=1 (screen)
C

```

```

C      IF ((WIN1.EQ.1) OR ((WIN2.EQ.1)) THEN
250  WRITE(' ') 'Anything to change ? (Y/N)'
      READ(' ',(A)) CHNG
      IF (CHNG.EQ.'Y') CHNG='Y'
      IF (CHNG.EQ.'N') CHNG='N'
      IF ((CHNG.NE.'Y') AND. (CHNG.NE.'N')) GOTO 250
      IF ((JOB.EQ.2) AND. ((WIN1.EQ.1)) WRITE('800)
      CALL GDAWK(1)
      CALL GCLWK(1)
      CALL GCLKS
      ENDIF
C
C      IF ((JOB.EQ.1) THEN
      WRITE(' ') 'CRT Display'
      IWIN2=0
      IWIN1=1
      ELSE
      WRITE(' ') 'PS File Creation'
      IWIN1=0
      IWIN2=1
      ENDIF
C
C      Open GKS Routines for General Case
C
C      CALL OPNGKS(2)
      DO 1 I=1,4
      ASFLST(I)=GINDIV
1 CONTINUE
      CALL SETASF(ASFLST)
      CALL OPNPW(SN', 0, IDEVS(JOB))
      CALL GAC(WX(1), 1)
      CALL GSWF(1, 1, XMIN, YXMAX, YVMIN, VYMAX)
      CALL GSWN(1, WXMIN, WXMAX, WYMIN, WYMAX)
      CALL GSELNT(1)
      CALL GSPLR(1, PL1, TYPE, LWIDTH, COL1)
      CALL GSTXFP(IFONT, 1)
      CALL GSLWSC(1.0)
C
C      IF (CHNG.EQ.'N') GOTO 350
C
C      select one of the following options for viewing
C
C      Eye point : location of eye in the same coordinate
C                  system where the object is displayed
C      Rotation angle: absolute angle with horizontal, rotate
C                  the picture not the object
C      Scale factor : scale the picture not the object
C      Hidden : ask whether hide those lines that are
C                  not supposed to be seen.
C      The status will remain until next
C      selection is made. The default is No

```

```

C      WRITE(' ') '> Eye Point: x,y,z'
      READ(' ') X1,Y1,Z1
      IF ((CHNG.EQ.'Y') AND. (JOB.EQ.2)) THEN
      WRITE(' ') '> Rotation angle'
      READ(' ') THETA
      STHETA=SIN(THETA'D2R)
      CTHETA=COS(THETA'D2R)
      WRITE(' ') '> Scale factor'
      READ(' ') FACTR
      WRITE(' ') '> Hidden ? (Y/N)'
      READ(' ',(A)) IH
      IF ((IHEQ.'Y') IH='Y'
      IF ((IHNE.'Y') IH='N'
      ELSE
      STHETA=0.0
      CTHETA=1.0
      FACTR=1.0
      IH='N'
      ENDIF
C
C      Determine the visible surfaces and their
C      normal vectors.
C
C      300 CONTINUE
      CALL NORMAL
C
C      Hidden Line Processing and Clipping
C
C      360 CONTINUE
      IF ((IHEQ.'Y') THEN
      DO 37 I=1,NP
37  IVIS(I)=0
      CALL HIDDEN
      ELSE
      IPL=IED
      ENDIF
C
C      350 IF ((JOB.EQ.2) THEN
      WRITE(' ') ' (PS File is prepared.)'
      ENDIF
C
C      ... clear workstation
      CALL GCLRWK(1,1)
C
C      DO 99 I=1,IPL
C
C      ... assign lines
      IF ((IHEQ.'Y') THEN
      CMESH(1,1)=YE(I,1)
      CMESH(2,1)=YE(I,1)
      CMESH(1,2)=XE(I,2)
      CMESH(2,2)=YE(I,2)
      ELSE

```

```

C      K=IEDGE(1,1)
C      L=IEDGE(1,2)
C      CMESH(1,1)=SX(K)
C      CMESH(2,1)=SY(K)
C      CMESH(1,2)=SX(L)
C      CMESH(2,2)=SY(L)
C      ENDIF
C
C      --- rotate lines
C      CALL ROTAT(CMESH,CTHETA,CTHETA)
C
C      --- scale lines
C      CMESH(1,1)=CMESH(1,1)*FACR
C      CMESH(2,1)=CMESH(2,1)*FACR
C      CMESH(1,2)=CMESH(1,2)*FACR
C      CMESH(2,2)=CMESH(2,2)*FACR
C
C      --- plot lines
C      CALL GPL(2,CMESH)
C      99 CONTINUE
C
C      --- write some texts
C      CALL GSTXAL(2,0)
C      TEXTPT(1)=0.
C      TEXTPT(2)=81.
C      CALL GSCHH(7)
C      CALL TEXT(TEXTPT,ITITLE)
C      TEXTPT(2)=87
C      CALL GSCHH(6)
C      CALL TEXT(TEXTPT,JNP)
C      TEXTPT(2)=92
C      CALL GSCHH(5)
C      CALL TEXT(TEXTPT,ILOGO)
C      TEXTPT(2)=97
C      CALL TEXT(TEXTPT,ILOGO1)
C      CALL GSTXAL(1,0)
C      TEXTPT(1)=20
C      TEXTPT(2)=65
C      CALL TEXT(TEXTPT,YRANG)
C      TEXTPT(2)=60
C      CALL TEXT(TEXTPT,XRANG)
C      TEXTPT(2)=55
C      CALL TEXT(TEXTPT,PKVAL)
C
C      --- draw the axes
C      CALL GSTXAL(2,0)
C      CALL AXES(ZCH,STHETA,CTHETA)
C
C      --- update workstation
C      CALL GUW(K,1,0)

```

```

C      --- modify if necessary
C      IF (L.OB.EQ.1) THEN
C        450 WRITE ('.',' ' >> [R]edo, or [Q]uit'
C        READ ('',(A5)) IPROC
C        IF ((IPROC.EQ.'Q') .OR. (IPROC.EQ.'q')) GOTO 400
C        IF ((IPROC.EQ.'R') .OR. (IPROC.EQ.'r')) THEN
C          select one of the following options for viewing
C
C          Eye point      : location of eye in the same coordinate
C                        : system where the object is displayed
C          Rotation angle: absolute angle with horizontal, rotate
C                        : the picture not the object
C          Scale factor  : scale the picture not the object
C          Hidden       : ask whether hide those lines that are
C                        : not supposed to be seen.
C                        : The status will remain until next
C                        : selection is made. The default is No
C
C        WRITE ('.',' ' >> [E]ye point'
C        WRITE ('.',' ' [R]otation'
C        WRITE ('.',' ' [S]cale'
C        WRITE ('.',' ' [H]idden'
C        READ ('',(A5)) IREDO
C        IF ((IREDO.EQ.'E') .OR. (IREDO.EQ.'e')) THEN
C          WRITE ('.',' ' > Eye Point x,y,z'
C          READ ('.',' ' X1,Y1,Z1
C          GOTO 300
C        ELSEIF ((IREDO.EQ.'R') .OR. (IREDO.EQ.'r')) THEN
C          WRITE ('.',' ' > Rotation angle'
C          READ ('.',' ' THETA
C          STHETA=SIN(THETA*D2R)
C          CTHETA=COS(THETA*D2R)
C          GOTO 350
C        ELSEIF ((IREDO.EQ.'S') .OR. (IREDO.EQ.'s')) THEN
C          WRITE ('.',' ' > Scale factor'
C          READ ('.',' ' FACTR
C          GOTO 350
C        ELSEIF ((IREDO.EQ.'H') .OR. (IREDO.EQ.'h')) THEN
C          480 WRITE ('.',' ' > Yes or No ?'
C          READ ('',(A1)) IH
C          IF ((IH.EQ.'Y') .IH='Y')
C            IF ((IH.EQ.'n') .IH='N')
C              IF ((IH.NE.'Y') .AND. (IH.NE.'N')) GOTO 480
C          GOTO 360
C        ELSE
C          GOTO 460
C        ENDIF
C      ENDIF
C      GOTO 450
C    ENDIF
C  GOTO 400

```

```

C
700 CONTINUE
C
C      --- finish up
C
      IF (WIN1 EQ 1) WRITE(*,800)
800 FORMAT(1X,' > Press Any Key within GKS Window.')
      CALL GDAWK(1)
      CALL GCLWK(1)
      CALL GCLKS
      STOP
      END
C-----
C
SUBROUTINE DATAIN(TITLE,ZCH)
C
C      This subroutine reads input data stored in file SSP.DAT
C
      PARAMETER(NPMAX=1000,NEMAX=500,NSMAX=1000,NNODE=4)
      COMMON /COORD/ X(NPMAX),Y(NPMAX),Z(NPMAX)
      COMMON /SURF/ ISUR,ISURF(NSMAX,5)
      COMMON /MEINF/ NP,NE
      REAL LWIDTH
      INTEGER PLI,COLI
      COMMON /GRAPH/ ORIGX,ORIGY,AXLEN,TXLEN,CHAHT,
& FONT,LTYPE,PLI,COLI,LWIDTH
      COMMON /OSCRP/ XMIN,XMAX,YMIN,YMAX,SMAX
C
      DIMENSION SS(NEMAX,NNODE),XELE(NEMAX,NNODE),
& YELE(NEMAX,NNODE),S(NPMAX),INP(NPMAX)
      CHARACTER*80 ITITLE,ZCH,TMP*10
C
      WRITE(*,*) '...DATAIN'
      OPEN(9,FILE='ssp.dat')
      READ(9, '(A)') ITITLE
1      WRITE(*,*) ' [P]ressure or [S]hear'
      READ(*, '(A)') ZCH
      IF (ZCHEQ.'P') ZCH='P'
      IF (ZCHEQ.'S') ZCH='S'
      IF ((ZCH.NE.'P') .AND. (ZCH.NE.'S')) GOTO 1
      IF (ZCHEQ.'P') ITITLE='Pressure distribution'
      IF (ZCHEQ.'S') ITITLE='Shear stress distribution'
      DMAX=0.
      SMAX=0.
      I=0
C
      read input data (stress distributions) produced by TTIM
C      in element-by-element (ExE) form
C
21      READ(9, '(A)',END=20) TMP
      if (tmp.EQ.'end') goto 20
      I=I+1
      READ(9,*) (ISURF(I,J),J=1,NNODE)
      DO 10 K=1,NNODE
C
      READ(9,*) XELE(I,K),ZDUMY,YELE(I,K),TEMPP,TEMPS
      IF (ZCHEQ.'P') THEN
        SS(I,K)=TEMPP
      ELSE
        SS(I,K)=TEMPS
      ENDIF
      TEMP=AMAX1(ABS(XELE(I,K)),ABS(YELE(I,K)))
      IF (TEMP.GT.DMAX) DMAX=TEMP
      TMPS=ABS(SS(I,K))
      IF (TMPS.GT.SMAX) SMAX=TMPS
10      CONTINUE
      GOTO 21
20      NE=I
      CLOSE(9)
C
C      read some parameters for plotting
C      WIND . window length in user unit
C      PLI . polyline index
C      LTYPE . polyline type
C      LWIDTH . polyline width scale factor
C      COLI . color index
C      IFONT . text font
C      ORIGX, ORIGY x, y coordinates of the axes in user unit
C      AXLEN . axis length
C      TXLEM . distance of text X, Y and Z from the origin
C      TXTLEN . text height for X, Y and Z in user unit
C      CHAHT . character height
C
      OPEN(19,FILE='ssgraph.dat')
      READ(19,*) WIND
      READ(19,*) PLI,LTYPE
      READ(19,*) LWIDTH,COLI
      READ(19,*) IFONT
      READ(19,*) ORIGX,ORIGY
      READ(19,*) AXLEN,TXLEN
      READ(19,*) CHAHT
      CLOSE(19)
C
C      renumber the nodal numbers
C
      IP=0
      DO 30 I=1,NE
      DO 31 J=1,NNODE
      DO 32 K=1,IP
      IF (ISURF(I,J) EQ.INP(K)) GOTO 31
32      CONTINUE
      IP=IP+1
      INP(IP)=ISURF(I,J)
31      CONTINUE
30      CONTINUE
C
C      convert coordinates and functions in ExE form to vectors

```

```

C
DO 40 I=1,NE
DO 41 J=1,NNODE
ITMP=ISURF(I,J)
DO 42 K=1,IP
IF (ITMP.EQ.INP(K)) THEN
ISURF(I,J)=K
X(K)=XELE(I,J)
Y(K)=YELE(I,J)
S(K)=SS(I,J)
ENDIF
42 CONTINUE
41 CONTINUE
ISURF(I,5)=ISURF(I,1)
40 CONTINUE
C
C make a mirror image of the above surface,
C the stress distributions are symmetrical about
C the plane of motion.
C
NP=IP
DO 60 I=1,NE
INE=NE+I
DO 61 J=1,(NNODE+1)
ISURF(INE,J)=ISURF(I,J)+NP
61 CONTINUE
60 CONTINUE
NE=NE*2
DO 70 I=1,NP
IPP=NP+I
X(IPP)=X(I)
Y(IPP)=-Y(I)
S(IPP)=S(I)
70 CONTINUE
NP=NP*2
C
C consider both sides of the surface visible
C
DO 50 I=1,NE
INE=NE+I
DO 51 J=1,(NNODE+1)
JREV=NNODE-J+2
ISURF(INE,JREV)=ISURF(I,J)
51 CONTINUE
50 CONTINUE
NE=NE*2
ISUR=NE
C
C since coordinates and functions have different units,
C scale the functions so that the diagram looks better
C
RATIO=0.8*DMAX/SMAX

```

```

DO 75 I=1,NP
Z(I)=S(I)*RATIO
75 CONTINUE
C
C calculate new origin
C
XMIN= 1.E+20
XMAX=-1.E+20
YMIN= 1.E+20
YMAX=-1.E+20
ZMIN= 1.E+20
ZMAX=-1.E+20
DO 80 I=1,NP
IF (X(I).LT.XMIN) XMIN=X(I)
IF (X(I).GT.XMAX) XMAX=X(I)
IF (Y(I).LT.YMIN) YMIN=Y(I)
IF (Y(I).GT.YMAX) YMAX=Y(I)
IF (Z(I).LT.ZMIN) ZMIN=Z(I)
IF (Z(I).GT.ZMAX) ZMAX=Z(I)
80 CONTINUE
XORIG=0.5*(XMIN+XMAX)
YORIG=0.5*(YMIN+YMAX)
ZORIG=0.5*(ZMIN+ZMAX)
C
C translate the coordinates
C
DO 90 I=1,NP
X(I)=X(I)-XORIG
Y(I)=Y(I)-YORIG
Z(I)=Z(I)-ZORIG
90 CONTINUE
C
C scale user coordinates to default coordinates.
C default window is 200x200, its origin is located
C at the center of the window
C leave 20% space as margin
C
SCALE=80.0/WIND
DO 100 I=1,NP
X(I)=X(I)*SCALE
Y(I)=Y(I)*SCALE
Z(I)=Z(I)*SCALE
100 CONTINUE
C
C scale the parameters for plotting the axes
C
SCALE=100.0/WIND
ORIGX=ORIGX*SCALE
ORIGY=ORIGY*SCALE
AXLEN=AXLEN*SCALE
TXLEN=TXLEN*SCALE
CHAHT=CHAHT*SCALE

```



```

C      RETURN
C      END
C-----SUBROUTINE NORMAL
C
C      This subroutine projects the coordinates onto
C      normal plane
C
C      PARAMETER(NPMA=1000, NEMAX=500, NSMAX=1000, NNODE=4)
C      COMMON /COORD/ X(NPMA), Y(NPMA), Z(NPMA)
C      COMMON /COORD1/ SX(NPMA), SY(NPMA), SZ(NPMA)
C      COMMON /MEIN/ NPNE
C      COMMON /VIEW/ X1,Y1,Z1,S1,C1,S2,C2
C      COMMON /SURF/ ISURF(NSMAX,5)
C      COMMON /PARAM/ D2R,R2D
C
C      WRITE(' ') ' NORMAL '
C
C      convert eye point (x,y,z) in Cartesian coordinate system
C      to spherical coordinate system
C
C      IF ((X1.EQ.0.) .AND. (Y1.GE.0.)) THEN
C        THETA=90.0*D2R
C      ELSEIF ((X1.EQ.0.) .AND. (Y1.LT.0.)) THEN
C        THETA=90.0*D2R
C      ELSEIF (X1.LT.0.) THEN
C        THETA=180.0*D2R+ATAN(Y1/X1)
C      ELSE
C        THETA=ATAN(Y1/X1)
C      ENDIF
C
C      VX2=SQRT(X1*X1+Y1*Y1)
C
C      IF (Z1.EQ.0.) THEN
C        PHAI=90.0*D2R
C      ELSEIF (Z1.GT.0.) THEN
C        PHAI=ATAN(VX2/Z1)
C      ELSEIF (Z1.LT.0.) THEN
C        PHAI=180.0*D2R+ATAN(VX2/Z1)
C      ENDIF
C
C      --- calculate the view distance
C      See Fig. 1
C
C      RHO=SQRT(X1*X1+Y1*Y1+Z1*Z1)
C
C      S1=SIN(THETA)
C      C1=COS(THETA)
C      S2=SIN(PHAI)
C      C2=COS(PHAI)
C
C      DO 30 I=1,NP
C
C      --- convert the coordinates to
C      the system X'Y'Z' at eye point
C
C      SX(I)=X(I)*S1 +Y(I)*C1
C      SY(I)=X(I)*C1*C2-Y(I)*S1*C2+Z(I)*S2
C      SZ(I)=X(I)*C1*S2-Y(I)*S1*S2-Z(I)*C2 + RHO
C
C      --- project the point onto
C      the view plane X''Y'' at origin
C
C      SX(I)=RHO*SX(I)/SZ(I)
C      SY(I)=RHO*SY(I)/SZ(I)
C      30 CONTINUE
C      RETURN
C      END
C-----SUBROUTINE CEDGE
C
C      This subroutine finds the visible edges or line segments
C
C      PARAMETER(NPMA=1000, NEMAX=500, NSMAX=1000, NNODE=4)
C      COMMON /COORD/ X(NPMA), Y(NPMA), Z(NPMA)
C      COMMON /COORD1/ SX(NPMA), SY(NPMA), SZ(NPMA)
C      COMMON /MEIN/ NPNE
C      COMMON /VIEW/ X1,Y1,Z1,S1,C1,S2,C2
C      COMMON /SURF/ ISURF(NSMAX,5)
C      COMMON /EDGES/ IED,IEDGE(NSMAX,2)
C      COMMON /PARAM/ D2R,R2D
C
C      WRITE(' ') ' CEDGE '
C
C      --- identify the visible edges
C
C      N=0
C      NEHLF=NE/2
C      DO 80 I=1,NEHLF
C        DO 82 J=1,NNODE
C          IE1=ISURF(I,J)
C          IE2=ISURF(I,J+1)
C          DO 84 K=1,N
C            JE1=IEDGE(K,1)
C            JE2=IEDGE(K,2)
C
C            if this line segment is the duplicate of
C            of a previously recorded one, then skip this one.
C
C            IF (((JE1.EQ.IE2) .AND. (JE2.EQ.IE1)) .OR.
C              &
C              ((JE1.EQ.IE1) .AND. (JE2.EQ.IE2))) GOTO 82
C            84 CONTINUE
C            N=N+1
C            IEDGE(N,1)=IE1
C            IEDGE(N,2)=IE2

```

```
      IEDGE(N,2)=IE2
82  CONTINUE
80  CONTINUE
C
      IED=N
      RETURN
      END
C
```



```

C      XC=0.0
      YC=0.0
      ZC=0.0
      DO 15 K=1,NNODE
        XC=XC + SX(ISURF(I,K))
        YC=YC + SY(ISURF(I,K))
        ZC=ZC + SZ(ISURF(I,K))
      15 CONTINUE
      XC=XC*0.25
      YC=YC*0.25
      ZC=ZC*0.25
C      divide the element into four triangles
C      using corner points and centroid
C
C      DO 20 K=1,NNODE
C        further divide each of the above triangles
C        into three triangle using two adjacent corner
C        points, the centroid and the point provided
C        calculate the areas of four triangles, including
C        the one has been divided
C
C        J1=ISURF(I,K 1)
C        J2=ISURF(I,K+1)
C
C        1 triangle J1-J2-Point
C          (the order of the points are clockwise)
C
C        S1=SX(J1)
C        R1=SY(J1)
C        S2=SX(J2)
C        R2=SY(J2)
C        S3=XT
C        R3=YT
C        CALL AREAC(S1,S2,S3,R1,R2,R3,AREA1)
C
C        2 triangle J2-Centroid Point
C
C        IF (AREA1.LT.1.0E-3) GOTO 20
C        S1=SX(J2)
C        R1=SY(J2)
C        S2=XC
C        R2=YC
C        S3=XT
C        R3=YT
C        CALL AREAC(S1,S2,S3,R1,R2,R3,AREA2)
C
C        3 triangle Centroid-J1-Point
C
C      IF (AREA1.LT.1.0E-3) GOTO 20
C      S1=XC
C      R1=YC
C      S2=SX(J1)
C      R2=SY(J1)
C      S3=XT
C      R3=YT
C      CALL AREAC(S1,S2,S3,R1,R2,R3,AREA3)
C
C      4 : triangle J1-J2-Centroid
C
C      IF (AREA2.LT.1.0E-3) GOTO 20
C      S1=SX(J1)
C      R1=SY(J1)
C      S2=SX(J2)
C      R2=SY(J2)
C      S3=XC
C      R3=YC
C      CALL AREAC(S1,S2,S3,R1,R2,R3,AREA)
C      calculate the Z-coordinate ZX that is in line with ZT
C      but on the plane of the triangle J1-J2-Centroid
C
C      ZX=(SZ(J1)*AREA1 + SZ(J2)*AREA2 + ZC*AREA3)/AREA
C
C      if ZX is less than ZT, it means ZT appears to be
C      behind the triangle J1-J2-Centroid, and it is
C      invisible
C
C      IF (ZX.LT.ZT) RETURN
C
C      20 CONTINUE
C      10 CONTINUE
C      if all ZX(s) are greater than ZT, then it is visible
C
C      IV=1
C      RETURN
C      END
C-----
C      SUBROUTINE AREAC(S1,S2,S3,R1,R2,R3,AREA)
C
C      This subroutine calculates the area of a triangle
C      defined by three points
C
C      (in fact AREA = 2 * area of the triangle)
C
C      AREA=S1*(R2-R3) + S2*(R3-R1) + S3*(R1-R2)
C
C      RETURN
C      END

```

```

C-----
SUBROUTINE CLIPSK(L,XV,YV,ZV,XT,YT,ZT)
C
C This subroutine clips the invisible lines
C
C
PARAMETER(NPMMAX=1000,NEMAX=500,NSMAX=1000,NNODE=4)
COMMON /COORD1/ SX(NPMMAX),SY(NPMMAX),SZ(NPMMAX)
COMMON /VISIV/ IVIS(NPMMAX)
C
C assign the coordinates according to the visibility of
C the point
C
IF (IVIS(K) EQ 1) THEN
  XV= SX(K)
  YV= SY(K)
  ZV= SZ(K)
  XH= SX(L)
  YH= SY(L)
  ZH= SZ(L)
  IV= IVIS(L)
ELSEIF (IVIS(L) EQ 1) THEN
  XV= SX(L)
  YV= SY(L)
  ZV= SZ(L)
  XH= SX(K)
  YH= SY(K)
  ZH= SZ(K)
  IV= IVIS(L)
ENDIF
C
C calculate the length of the line segment
C
C
V1=(XH-XV)
V2=(YH-YV)
V3=(ZH-ZV)
XT=XV
YT=YV
ZT=ZV
C
C clip the line segment by half, then check
C the visibility by calling VISIBL, and clip
C according to the new visibility of the point,
C which is stored in IV (line may be shortened or
C elongated) do this 12 times so that the clipping
C is good enough.
C
DO 10 KR=1,12
  V1=V1*0.5
  V2=V2*0.5
  V3=V3*0.5
  XT=XT + FLOAT(IV)*V1
  YT=YT + FLOAT(IV)*V2

```

```

  ZT=ZT + FLOAT(IV)*V3
  CALL VISIBL(XT,YT,ZT,IV)
10 CONTINUE
RETURN
END
C-----
SUBROUTINE AXES(ZCH,STHETA,CTHETA)
C
C This subroutine plots the coordinate system in use
C
COMMON /VIEW/ X1,Y1,Z1,S1,C1,S2,C2
REAL LWIDTH
INTEGER PLI,COLI
COMMON /GRAPH/ ORIGX,ORIGY,AXLEN,TXLEN,CHAHT,
& FONT,LTYPE,PLI,COLI,LWIDTH
DIMENSION AX(3),AY(3),AZ(3),X(3),Y(3),CMESH(2,2)
DIMENSION TX(3),TY(3),TZ(3),XX(3),YY(3),CC(2)
CHARACTER*80 STR(3),ZCH
C
DO 2 I=1,3
  AX(I)=0.0
  AY(I)=0.0
  AZ(I)=0.0
  TX(I)=0.0
  TY(I)=0.0
  TZ(I)=0.0
2 CONTINUE
C
C set axis lengths
C
C
AX(1)=AXLEN
AY(2)=AXLEN
AZ(3)=AXLEN
C
C set the distances where X, Y and Z are to be plotted
C
C
TX(1)=TXLEN
TY(2)=TXLEN
TZ(3)=TXLEN
C
C assign the characters
C
C
STR(1)='X'
STR(2)='Y'
STR(3)='ZCH'
C
C project the axes
C
DO 1 I=1,3
  X(I)=-AX(I)*S1 + AY(I)*C1
  Y(I)=-AX(I)*C1*CC2 - AY(I)*S1*CC2 + AZ(I)*S2
  XX(I)=-TX(I)*S1 + TY(I)*C1

```

```

YY(I)=TX(I)*C1*C2 - TY(I)*S1*C2 + TZ(I)*S2
1 CONTINUE
C
C rotate, shift and plot the axes
C
DO 9 I=1,3
  CMESH(1,1)=0.0
  CMESH(2,1)=0.0
  CMESH(1,2)=X(I)
  CMESH(2,2)=Y(I)
C
  CALL ROTAT(CMESH,STHETA,CTHETA)
C
  CMESH(1,1)=CMESH(1,1) + ORIGX
  CMESH(2,1)=CMESH(2,1) + ORIGY
  CMESH(1,2)=CMESH(1,2) + ORIGX
  CMESH(2,2)=CMESH(2,2) + ORIGY
  CALL GPL(2,CMESH)
9 CONTINUE
C
C rotate, shift and plot the X,Y and Z
C
CALL GSTXAL(2,3)
CALL GSCHH(CHAHT)
DO 5 I=1,3
  CC(1)=XX(I)*CTHETA + YY(I)*STHETA
  CC(2)=XX(I)*STHETA + YY(I)*CTHETA
  CC(1)=CC(1) + ORIGX
  CC(2)=CC(2) + ORIGY
  CALL TEXT(CC,STR(I))
5 CONTINUE
RETURN
END
C .....
SUBROUTINE ROTAT(CMESH,STHETA,CTHETA)
C
C This subroutine performs coordinate rotation in a plane
C
  DIMENSION CMESH(2,2) COORD(2,2)
C
  DO 1 J=1,2
    COORD(1,J)= CMESH(1,J)*CTHETA + CMESH(2,J)*STHETA
    COORD(2,J)= CMESH(1,J)*STHETA + CMESH(2,J)*CTHETA
1 CONTINUE
C
  DO 2 I=1,2
    DO 2 J=1,2
      CMESH(I,J)=COORD(1,J)
2 CONTINUE
RETURN
END
C

```

```

C.....
C      SV
C      .. SV (Section View) plots the view of pressure or
C      stress distribution for a given section (x,y,z)
C      Library Used: GKS
C      by Songtao Chen
C      Department of Mechanical and Aerospace Engineering
C      Carleton University
C      (Ver 1.0, Feb., 1993)
C.....
C      To understand this software, knowledge of 2D computer graphics
C      and GKS libraries are essential
C.....
C      INCLUDE 'usr/local/lib/gks/gksddl.f'
C      PARAMETER(NPMAX=1000,NEMAX=600,NSMAX=1000,NNODE=4)
C      IMPLICIT REAL*8 (A-H,O-Z)
C      COMMON /COORD/ XU(NPMAX), YU(NPMAX), ZU(NPMAX), SS(NPMAX)
C      COMMON /SURF/ NECT,ISURF(NEMAX,NNODE)
C      COMMON /COORD1/ XT(NPMAX), YT(NPMAX), ZT(NPMAX)
C      COMMON /TRSRF/ NETR,NNODE(NEMAX,NNODE)
C      COMMON /MEINF/ NP,NE
C      COMMON /PARAM/ D2R,R2D,PI
C      COMMON /PLOTS/ XE(NPMAX),YE(NPMAX),PXE(NPMAX),PYE(NPMAX)
C      REAL*4 LWIDTH
C      INTEGER PLI,COLI
C      COMMON /GRAPH/ ORIGX,ORIGY,AXLEN,TXLEN,CHAHT,
C      & IFONT,LTYPE,PLI,COLI,LWIDTH
C
C      REAL*4 CMESH(2,2),PMESH(2,2),RMESH(2,2),TMSH(2,2),
C      & TEXTPT(2)
C      REAL*4 WXMN,WXMAX,WYMN,WYMAX,VXMN,VXMAX,VYMN,VYMAX
C      REAL*4 XTC,YTC
C      REAL*8 PSVAL(NPMAX),PSVAL(NPMAX),XE0(NPMAX),YE0(NPMAX)
C      REAL*8 XR(NPMAX),YR(NPMAX),XR0(NPMAX),YR0(NPMAX)
C      CHARACTER*80 ILOGO,ILOGO1,ZCH
C      CHARACTER*80 IDEVS(2)
C      CHARACTER*80 ITITLE,PKVAL
C      CHARACTER*1 IPROC,IPEDO,CHNG,IV
C      INTEGER ASFLST(4)
C
C      there are two output devices, screen and postscript file
C
C      IDEVS(1)='screen'
C      IDEVS(2)='gks2ps'
C      XTC=20.
C      YTC=55.
C
C      the following text will appear on all the diagrams
C      produced by this program
C.....
ILOGO='Transport Technology Research Laboratory'
ILOGO1='Carleton University, Ottawa, CANADA'
PI=4 D0'DATAN(1 D0)
D2R=PI/180.
R2D=180 D/PI
C      print headline and read input data
C
C      WRITE(' ','') Welcome to Section View (SV)
C      CALL DATIN(ITITLE,ZCH)
C
C      Set Window size (default is 200x200).
C      origin is set at the center of the window
C      Set Viewport size as default
C
C      WXMN=100 D
C      WXMAX=100 D
C      WYMN=100 D
C      WYMAX=100 D
C      VXMN=0 D
C      VXMAX=1 D
C      VYMN=0 D
C      VYMAX=1 D
C
C      Menu of Job
C
C      IWIN1=0
C      IWIN2=0
C      CHNG='Y'
C
C      400 WRITE(' ','') 3D Graphics Menu
C      WRITE(' ','') 1 CRT 2 PS File 3 Exit
C      WRITE(' ','') Your job:
C      READ(' ','') IJOB
C      IF ((IJOB EQ 1) .OR. (IJOB EQ 2)) GOTO 500
C      IF ((IJOB EQ 3) .OR. (IJOB EQ 0)) GOTO 700
C      GOTO 400
C
C      CRT Display and Postscript File Output
C
C      500 CONTINUE
C
C      Open GKS Routines for IJOB=2 (postscript file)
C      after IJOB=1 (screen)
C
C      IF ((IWIN1 EQ 1) .OR. (IWIN2 EQ 1)) THEN
C      CHNG='N'
C      IF ((IJOB EQ 2) .AND. (IWIN1 EQ 1)) WRITE(' ',800)
C      CALL GDAWK(1)
C      CALL GCLWK(1)
C      CALL GCLKS
C      ENDIF

```

```

C      that are intercepted by the plane at x
C
C      KPT=0
C
C      set xi at .1, since 4-node element is used, only coordinates
C      at edges are necessary to define the line formed by the view
C      plane and the element.
C
C      XI=.1;DO
C
C      loop through all the elements
C
C      DO 11 I=1,NECNT
C
C      call subroutine LOCATE to find the coordinates and
C      normal pressure or shear stress at edges (JOB = 1)
C
C
C      IF (IV EQ X') THEN
C          CALL LOCATE(ZU,YU,XU,SS,XYZ,XI,CX,CY,PS,I,ANS,1,IV)
C      ELSEIF (IV EQ Y') THEN
C          CALL LOCATE(XU,ZU,YU,SS,XYZ,XI,CY,CX,PS,I,ANS,1,IV)
C      ELSE
C          CALL LOCATE(YU,XU,ZU,SS,XYZ,XI,CY,CX,PS,I,ANS,1,IV)
C      ENDIF
C
C      if the operation is successful (IANS = 1), then store
C      them
C
C      CX    x-coordinate in the view plane
C      CY    y-coordinate in the view plane
C      PS    pressure or shear stress
C
C
C      IF (IANSEQ 1) THEN
C          KPT=KPT+1
C          IF ((KPT EQ 1) AND (IV NE Y')) THEN
C              IFST=1
C              KPT=KPT+1
C          ENDIF
C          XEQ(KPT)=CX
C          YEQ(KPT)=CY
C          PSVALQ(KPT)=PS
C      ENDIF
C
C      1* CONTINUE
C
C      IF (IV EQ Z') THEN
C          NETR=NE
C      ELSE
C          NETR=2*NE-3
C      ENDIF
C      KPW=0
C
C      DO 21 I=1,NETR

```



```

FACTR=1.0
XS=0.0
YS=0.0
IF=1W
ENDIF

C
C translate distribution and contour coordinates
C by XS and YS, respectively
C
370 DO 16 I=1,KPT
C
C coordinates for distributions
C
XE(I)=XEQ(I)+XS
YE(I)=YEQ(I)+YS
16 CONTINUE
DO 23 I=1,KPW
C
C coordinates for contour
C
XR(I)=XRQ(I)+XS
YR(I)=YRQ(I)+YS
23 CONTINUE
C
C scale distribution and contour
C
DO 15 I=1,KPT
PSVAL(I)=PSVALQ(I)*SCALE
XE(I)=XE(I)*FACTR
YE(I)=YE(I)*FACTR
15 CONTINUE
DO 24 I=1,KPW
XR(I)=XR(I)*FACTR
YR(I)=YR(I)*FACTR
24 CONTINUE
C
C get the maximum pressure or shear stress
C
XMI= 1 D+10
XMA=-1 D+10
YMI= 1 D+10
YMA=-1 D+10
SMA=-1 D+10
DO 13 I=1,KPT
IF (XE(I) GT XMA) XMA=XE(I)
IF (XE(I) LT XMI) XMI=XE(I)
IF (YE(I) GT YMA) YMA=YE(I)
IF (YE(I) LT YMI) YMI=YE(I)
IF (PSVALQ(I) GT SMA) SMA=PSVALQ(I)
13 CONTINUE
C
C IF (ZCHEQ P) THEN
WRITE(PKVAL,1230) SMA
1230 FORMAT('Peak pressure: ',F9.2,' kPa')
ELSE
WRITE(PKVAL,1231) SMA
1231 FORMAT('Peak shear: ',F9.2,' kPa')
ENDIF
C
C calculate the envelope of the distribution
C
XC=0.5D0*(XMA+XMI)
YC=0.5D0*(YMA+YMI)
DO 12 I=1,KPT
CALL PREP(PSVAL,XC,YC,I,KPT,IV)
12 CONTINUE
IPL=KPT-1
350 IF (JOB.EQ.2) THEN
WRITE ('.') ' (PS File is prepared.)'
ENDIF
C
C --- clear workstation
C CALL GCLRWK(1,1)
C
DO 99 I=1,IPL
C
C plot distribution
C --- assign base lines
CMESH(1,1)=XE(I)
CMESH(2,1)=YE(I)
CMESH(1,2)=XE(I+1)
CMESH(2,2)=YE(I+1)
C
C --- assign envelope lines
PMESH(1,1)=PXE(I)
PMESH(2,1)=PYE(I)
PMESH(1,2)=PXE(I+1)
PMESH(2,2)=PYE(I+1)
C
C --- assign radial lines
RMESH(1,1)=XE(I)
RMESH(2,1)=YE(I)
RMESH(1,2)=PXE(I)
RMESH(2,2)=PYE(I)
C
C --- plot lines
CALL GPL(2,CMESH)
CALL GPL(2,PMESH)
CALL GPL(2,RMESH)
99 CONTINUE
C
C plot contour
C
DO 25 I=1,KPW+1
C --- assign contour lines
TMESH(1,1)=XR(I)
TMESH(2,1)=YR(I)

```

```

      TMSH(1,2)=XR(1+1)
      TMSH(2,2)=YR(1+1)
      --- plot contour
      CALL GPL(2,TMSH)
25 CONTINUE
C
C      --- write some texts
      CALL GSTXAL(2,0)
      TEXTPT(1)=0.
      TEXTPT(2)=81.
      CALL GSCHH(7)
      CALL TEXT(1,TEXTPT,ITITLE)
      TEXTPT(2)=92.
      CALL GSCHH(5)
      CALL TEXT(1,TEXTPT,ILOGO)
      TEXTPT(2)=97
      CALL TEXT(1,TEXTPT,ILOGO1)
      CALL GSTXAL(1,0)
      TEXTPT(1)=XTC
      TEXTPT(2)=YTC
      CALL TEXT(1,TEXTPT,PKVAL)
C
C      --- draw the axes
      CALL GSTXAL(2,0)
      --- update workstation
      CALL GUWK(1,0)
C
C      --- modify if necessary
C
C      You can select different view planes as discussed
C      earlier
C
C      You can scale the diagram by providing two scale
C      factors for distribution and contour
C
C      You can move the diagram by providing the amount
C      of displacement (in user units) required in
C      each direction
C
C      You can move the text by providing the new location
C
C      IF (IJOB.EQ.1) THEN
450  WRITE ('.') ' >> [R]edo, or [Q]uit'
      READ ('.(A1)') IPROC
      IF ((IPROC.EQ.'Q') .OR. (IPROC.EQ.'q')) GOTO 400
      IF ((IPROC.EQ.'R') .OR. (IPROC.EQ.'r')) THEN
460    WRITE ('.') ' [Scale'
        WRITE ('.') ' [Diagram move'
        WRITE ('.') ' [Text move'
        WRITE ('.') ' [View plane'
        READ ('.(A1)') IREDO

```

```

      IF ((IREDO.EQ.'V') .OR. (IREDO.EQ.'v')) THEN
        GOTO 250
      ELSEIF ((IREDO.EQ.'S') .OR. (IREDO.EQ.'s')) THEN
        WRITE ('.') ' > Scale factor?'
        READ ('.') FACTR,SCALE
        GOTO 370
      ELSEIF ((IREDO.EQ.'D') .OR. (IREDO.EQ.'d')) THEN
        WRITE ('.') ' > Xs & Ys '
        READ ('.') XS,YS
        GOTO 370
      ELSEIF ((IREDO.EQ.'T') .OR. (IREDO.EQ.'t')) THEN
        WRITE ('.') ' > Xtc & Ytc '
        READ ('.') XTC,YTC
        GOTO 350
      ELSE
        GOTO 460
      ENDIF
    ENDIF
    GOTO 450
  ENDIF
  GOTO 400
C
C 700 CONTINUE
C      --- finish up
      IF ((WIN1.EQ.1) WRITE ('.800)
      800 FORMAT(1X,' > Press Any Key within GKS Window.')
      CALL GDAWK(1)
      CALL GCLWK(1)
      CALL GCLKS
      STOP
      END
C-----
SUBROUTINE PREP(PSVAL,XC,YC,I,KPT,IV)
C
C This subroutine calculates the envelope of the distribution
C
C PARAMETER(NPMAX=1000,NEMAX=630,NSMAX=1000,NNODE=4)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 PSVAL(NPMAX)
COMMON /PLOTS/ XE(NPMAX),YE(NPMAX),PXE(NPMAX),PYE(NPMAX)
COMMON /PARAM/ D2R,R2D,P1
CHARACTER*1 IV
C
C IF ((I.GT.1) .AND. (I.LT.KPT)) THEN
C
C calculate the slopes and the angles
C
C PK1=(YE(I)-YE(I-1))/(XE(I)-XE(I-1))
C PK2=(YE(I+1)-YE(I))/(XE(I+1)-XE(I))
C ALF1=DATAN(PK1)
C ALF2=DATAN(PK2)
C IF ((ALF1*ALF2.LT.0.D0) .AND.

```

```

& (DABS(ALF1)+DABS(ALF2) GT. 0.5D0*PI)) T1 = N
  ALFA=0.5D0*(ALF1+ALF2)
ELSE
  ALFA=0.5D0*(P1+ALF1+ALF2)
ENDIF

C
C      two possible directions of projection
C      inward or outward
C
P1=XE(I)-PSVAL(I)*DCOS(ALFA)
P2=YE(I)-PSVAL(I)*DSIN(ALFA)
Q1=XE(I)+PSVAL(I)*DCOS(ALFA)
Q2=YE(I)+PSVAL(I)*DSIN(ALFA)

C      calculate the distance between the point of application
C      and the centroid, and determine the outward direction
C
D1=(P1-XC)*(P1-XC)+(P2-YC)*(P2-YC)
D2=(Q1-XC)*(Q1-XC)+(Q2-YC)*(Q2-YC)
IF (D2.GT.D1) THEN

C
C      Q1, Q2 represent outward point
C
PXE(I)=Q1
PYE(I)=Q2
ELSE

C
C      P1, P2 represent outward point
C
PXE(I)=P1
PYE(I)=P2
ENDIF
ELSE
  PXE(I)=XE(I)
  PYE(I)=YE(I)
ENDIF

C      RETURN
C      END

C.....
SUBROUTINE DATIN(TITLE,ZCH)

C      This subroutine reads input data stored in file SV.DAT
C      reads FEM mesh in GENE3D.DAT
C      reads final coordinates in INTERP.OUT
C
PARAMETER(NPMAX=1000,NEMAX=600,NSMAX=1000,NNODE=4)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /COORD/ X(NPMAX),Y(NPMAX),Z(NPMAX),S(NPMAX)
COMMON /COORD1/ XT(NPMAX),YT(NPMAX),ZT(NPMAX)
COMMON /TRSPF/ NETR,NODE(NEMAX,NNODE)
COMMON /NEINF/ NP,NE

COMMON /SURF/ NECNT,ISURF(NEMAX,NNODE)
REAL*4 LWIDTH
INTEGER PLI,COLI
COMMON /GRAPH/ ORIGX,ORIGY,AXLEN,TXLEN,CHART,
& IFONT,LTYPE,PLI,COLI,LWIDTH
C
DIMENSION SS(NEMAX,NNODE),XE(NEMAX,NNODE),YE(NEMAX,NNODE),
& ZELE(NEMAX,NNODE),INP(NPMAX)
CHARACTER*80 ITITLE,ZCH,TMP*10

C      WRITE (':') '...DATA:!'
C      OPEN (9,FILE='sv.dat')
C      READ (9,'A') ITITLE
1 WRITE (':',' (P)ressure or (S)hear'
  READ (':','A') ZCH
  IF (ZCH.EQ.'P') ZCH='P'
  IF (ZCH.EQ.'S') ZCH='S'
  IF ((ZCH.NE.'P') .AND. (ZCH.NE.'S')) GOTO 1
  IF (ZCH.EQ.'P') ITITLE='Pressure distribution'
  IF (ZCH.EQ.'S') ITITLE='Shear stress distribution'
  I=0
  C      read input data (stress distributions) produced by TTIM
  C      in element by element (ExE) form
  C
21 READ(9,'(A)',END=20) TMP
  IF (TMP.EQ.'end') goto 20
  I=I+1
  READ(9,'') (ISURF(I,J),J=1,NNODE)
  DO 10 K=1,NNODE
    READ(9,'') XE(I,K),YE(I,K),ZELE(I,K),TEMPP,TEMPS
    IF (ZCH.EQ.'P') THEN
      SS(I,K)=TEMPP
    ELSE
      SS(I,K)=TEMPS
    ENDIF
  10 CONTINUE
  GOTO 21
20 NE=I
  NECNT=NE
  CLOSE(9)

C
C      read FEM nodal numbers
C
OPEN(9,FILE='gene3d.in')
READ(9,'(I)')
READ(9,'') NP,NE
DO 6 I=1,NP
  READ(9,'')
6 CONTINUE
DO 2 I=1,NE
  READ(9,'') IDUMY,(LNODE(I,J),J=1,NNODE)

```

```

2  CONTINUE
CLOSE(9)
C
C read the mesh for deformed tre
C
OPEN(9,FILE='interp.out')
READ(9,(4(I)))
NLP=(NP/15)/2
K=0
KT=0
DO 3 I=1,NLP
DO 4 J=1,15
K=K+1
READ(9,*) IDUMY,XT(K),YT(K),ZT(K)
4  CONTINUE
DO 5 J=1,15
READ(9,*)
5  CONTINUE
3  CONTINUE
CLOSE(9)
C
C read some parameters for plotting
C WIND : window length in user unit
C PLI : polyline index
C LTYPE : polyline type
C LWIDTH : polyline width scale factor
C COLI : color index
C IFONT : text font
C ORIGX, ORIGY : x, y coordinates of the axes in user unit
C AXLEN : axis length
C TXLEM : distance of text X, Y and Z from the origin
C TXTLEN : text height for X, Y and Z in user unit
C CHAHT : character height
C
OPEN(19,FILE='ssgraph.dat')
READ(19,*) WIND
READ(19,*) PLLTYPE
READ(19,*) LWIDTH,COLI
READ(19,*) IFONT
READ(19,*) ORIGX,ORIGY
READ(19,*) AXLEN,TXLEN
READ(19,*) CHAHT
CLOSE(19)
C
C renumber the nodal numbers
C
IP=0
DO 30 I=1,NE
DO 31 J=1,NNODE
DO 32 K=1,IP
IF (ISURF(I,J).EQ.INP(K)) GOTO 31
32  CONTINUE
C
IP=IP+1
INP(IP)=ISURF(I,J)
31  CONTINUE
30  CONTINUE
NPT=IP
C
C assign coordinates and functions in an ExE form
C to vectors
C
DO 40 I=1,NECNT
DO 41 J=1,NNODE
ITMP=ISURF(I,J)
DO 42 K=1,NPT
IF (ITMP.EQ.INP(K)) THEN
ISURF(I,J)=K
X(K)=XELE(I,J)
Y(K)=YELE(I,J)
Z(K)=ZELE(I,J)
S(K)=SS(I,J)
ENDIF
42  CONTINUE
41  CONTINUE
40  CONTINUE
NPT=NPT
C
NESECT=NE/14
DO 50 I=1,NESECT
DO 51 J=1,14
IJ=(I-1)*14 + J
IM=(I-1)*15
IN=I*15
LNODE(I,1)=LNODE(I,1)-IM
LNODE(I,2)=LNODE(I,2)-IM
IF (ILT,NESECT) THEN
LNODE(I,3)=LNODE(I,3)-IN
LNODE(I,4)=LNODE(I,4)-IN
ENDIF
51  CONTINUE
50  CONTINUE
C
RETURN
END
C-----
SUBROUTINE LOCATE(XU,YU,ZU,SS,Z0,XI,CX,CY,PS,IE,IANS,JOB,IV)
C
C This subroutine locates the contact element in which the given
C point resides, and completes the coordinates of the given point.
C
C given Y and Z, search for X.
C given Z and X, search for Y.
C given X and Y, search for Z.
C

```

```

C
PARAMETER (NPMAX=1000,NEMAX=600,NNODE=4)
IMPLICIT REAL*8 (A-H,O-Z)

C
REAL*8 A(NNODE)
REAL*8 XU(NPMAX),YU(NPMAX),ZU(NPMAX),SS(NPMAX)
CHARACTER*1 IV

C
C since xi is already set either -1 or 1, the object is to find
C the corresponding eta.
C
C
C
C IANS=0
CALL CALABC(IE,ZUA,JOB)
IF (IV.EQ.'X') THEN

C
C if X view plane is selected, calculate the eta
C
C
C TMP=A(2)+A(4)*XI
C IF (TMP.EQ.0.D0) GOTO 1
C ETA=(Z0-A(1)+A(3)*XI)/TMP
C ELSEIF (IV.EQ.'Y') THEN

C
C if Y view plane is selected (two possible roots)
C
C
C calculate the first eta and if it is valid
C get the pressure or shear stress (IV = 2)
C
C
C TMP=A(2)+A(4)*XI
C IF (TMP.EQ.0.D0) GOTO 1
C ETA=(Z0-A(1)+A(3)*XI)/TMP
C CALL ROUND(ETA)
C IF (DABS(ETA) LE 1.D0) THEN
C CALL CALXPS(XI,ETA,XU,YU,ZU,CX,CY,SS,PS,IE,2,JOB,IV)
C IANS=1
C GOTO 2
C ENDIF

C
C get the second eta
C
C
C TMP=A(3)+A(4)*XI
C IF (TMP.EQ.0.D0) GOTO 1
C ETA=(Z0-A(1)+A(2)*XI)/TMP
C ELSEIF (IV.EQ.'Z') THEN
C TMP=A(3)+A(4)*XI
C IF (TMP.EQ.0.D0) GOTO 1
C ETA=(Z0-A(1)+A(2)*XI)/TMP
C ENDIF

C
C if eta is valid, get the pressure or shear stress (IV = 1)
C
C
C CALL ROUND(ETA)
C IF (DABS(ETA) LE 1.D0) THEN
C CALL CALXPS(XI,ETA,XU,YU,ZU,CX,CY,SS,PS,IE,1,JOB,IV)

```

```

C
IANS=1
GOTO 2
ENDIF

1 CONTINUE
2 CONTINUE
C
RETURN
END

C-----
SUBROUTINE CALABC(IE,X,A,JOB)
C
C if the element is found, then calculate the coefficients for
C solving ETA.
C
C
C PARAMETER(NEMAX=600,NNODE=4)
C COMMON /SURFS/ NECNT,ISURF(NEMAX,NNODE)
C COMMON /TRSRF/ NETR,LNODE(NEMAX,NNODE)
C REAL*8 X(*)/A( )
C
C JOB : 1 -> distribution
C JOB : 2 -> contour
C
C IF (JOB.EQ.1) THEN
C I1=ISURF(IE,1)
C I2=ISURF(IE,2)
C I3=ISURF(IE,3)
C I4=ISURF(IE,4)
C ELSE
C I1=LNODE(IE,1)
C I2=LNODE(IE,2)
C I3=LNODE(IE,3)
C I4=LNODE(IE,4)
C ENDIF
C
C A(1)= 0.25D0*(X(I1) + X(I2) + X(I3) + X(I4))
C A(2)= -0.25D0*(X(I1) + X(I2) - X(I3) - X(I4))
C A(3)= -0.25D0*(X(I1) - X(I2) + X(I3) + X(I4))
C A(4)= 0.25D0*(X(I1) - X(I2) + X(I3) - X(I4))
C
C
C RETURN
C END
C-----
SUBROUTINE CALXPS(XI,ETA,X,Y,Z,CX,CY,SS,PS,IE,IV,JOB,IV)
C
C This subroutine calculates the other coordinate
C after ETA has been found in LOCATE
C
C
C PARAMETER (NEMAX=600,NNODE=4,LCOOD=2,NDDOF=3)
C IMPLICIT REAL*8 (A-H,O-Z)
C COMMON /SURFS/ NECNT,ISURF(NEMAX,NNODE)
C COMMON /TRSRF/ NETR,LNODE(NEMAX,NNODE)

```

```

C
C CHARACTER*1 N
C
C REAL*8 X('),Y('),Z('),SS(')
C REAL*8 FUNC(4),DER(2,4),COORD(4,3),DERIV(2,3)
C
C get 2D shape function and their derivatives
C
C CALL FMIIN2(3,DER2,LCCOD,FUNC,XI,ETA)
C CX=0.D0
C CY=0.D0
C PS=0.D0
C KJ=0
C
C do interpolation using shape function according to
C the view plane selected and the job requested
C
C IF (IV.EQ.'X') KJ=1
C IF ((IV.EQ.'Y') .AND. (IV.EQ.2)) THEN
C
C re-arrange the shape function
C
C KJ=3
C TEMP=FUNC(1)
C FUNC(1)=FUNC(4)
C FUNC(4)=TEMP
C TEMP=FUNC(2)
C FUNC(2)=FUNC(3)
C FUNC(3)=TEMP
C
C
C interpolate the coordinate according to the job
C
C DO 1 J=1,NNODE
C IF (JOB.EQ.1) THEN
C K=ISURF(IE,J)
C ELSE
C K=LNODE(IE,J)
C
C JI=J+KJ
C IF (JI.GT.NNODE) JI=JI-NNODE
C CX=CX+FUNC(JI)*X(K)
C CY=CY+FUNC(JI)*Y(K)
C IF (JOB.EQ.1) PS=PS+FUNC(JI)*SS(K)
C 1 CONTINUE
C
C calculate the projection of the pressure or shear stress
C on the view plane
C
C IF (JOB.EQ.1) THEN
C
C assign the coordinates
C
C
C DO 10 I=1,NNODE
C K=ISURF(IE,I)
C COORD(1,1)=X(K)
C COORD(1,2)=Y(K)
C COORD(1,3)=Z(K)
C 10 CONTINUE
C
C get the derivatives
C
C DO 2 I=1,LCCOD
C DO 2 J=1,NDDOF
C DERIV(I,J)=0.D0
C DO 3 K=1,NNODE
C DERIV(I,J)=DERIV(I,J)+DER2(I,K)*COORD(K,I)
C 3 CONTINUE
C 2 CONTINUE
C
C get the normal vector
C
C A1=DERIV(1,2)*DERIV(2,3)-DERIV(1,3)*DERIV(2,2)
C A2=DERIV(1,3)*DERIV(2,1)-DERIV(1,1)*DERIV(2,3)
C A3=DERIV(1,1)*DERIV(2,2)-DERIV(1,2)*DERIV(2,1)
C
C get the norm
C
C AM=DSQRT(A1*A1+A2*A2+A3*A3)
C
C calculate the projection according to the view plane
C
C IF (IV.EQ.'X') THEN
C CST=DSQRT(A2*A2+A3*A3)/AM
C ELSEIF (IV.EQ.'Y') THEN
C CST=DSQRT(A1*A1+A3*A3)/AM
C ELSE
C CST=DSQRT(A1*A1+A2*A2)/AM
C
C ENDIF
C PS=PS*CST
C ENDIF
C
C RETURN
C END
C-----
C SUBROUTINE FMIIN2(IJOB,DER2,IDER2,FUN2,XI,ETA)
C
C This subroutine calculates 2D shape functions and derivatives
C
C Job=1 : calculate shape functions only
C Job=2 : calculate derivatives of shape functions only
C
C otherwise : calculate both shape functions and their
C derivatives
C
C

```

```

C
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 DER2(IDER2,*),FUN2(*)
C
XIM =1.D0-XI
ETAM=1.D0-ETA
XIP =1.D0+XI
ETAP=1.D0+ETA
IF (IJOB.EQ.2) GOTO 10
C
-- Shape functions
FUN2(1)= 0.25D0*XIM*ETAM
FUN2(2)= 0.25D0*XIM*ETAP
FUN2(3)= 0.25D0*XIP*ETAP
FUN2(4)= 0.25D0*XIP*ETAM
IF (IJOB.EQ.1) RETURN
C
-- Derivatives of gzi
10 DER2(1,1)=-0.25D0*ETAM
DER2(1,2)=-0.25D0*ETAP
DER2(1,3)= 0.25D0*ETAP
DER2(1,4)= 0.25D0*ETAM
C
-- Derivatives of eia
DER2(2,1)=-0.25D0*XIM
DER2(2,2)= 0.25D0*XIM
DER2(2,3)= 0.25D0*XIP
DER2(2,4)=-0.25D0*XIP
C
RETURN
END
C
SUBROUTINE ROUND(X)
C
C due to numerical error, X may be slightly off what it is supposed
C to be. if X is very close to 1, let it be 1.
C
REAL*8 X
C
IF (DABS(1.D0-DABS(X)) LT 1.0D-8) X=X/DABS(X)
RETURN
END
C
SUBROUTINE SORT(N,XE,YE,PSVAL,JOB)
C
C This subroutine sorts out the stored data
C it sorts out YE first, and then based on that
C sorts out XE and PSVAL
C
PARAMETER(NPMAX=1000)
REAL*8 XE(NPMAX),YE(NPMAX),PSVAL(NPMAX),TEMP(NPMAX)
INTEGER INDX(NPMAX)
C
DO 11 J=1,N
INDX(J)=J
11 CONTINUE
C
IF (N.EQ.1) RETURN
L=N/2+1
IR=N
10 CONTINUE
IF (L.GT.1) THEN
L=L-1
INDXT=INDX(L)
Q=YE(INDXT)
ELSE
INDXT=INDX(IR)
Q=YE(INDXT)
INDX(IR)=INDX(1)
IR=IR-1
IF (IREQ.1) THEN
INDX(1)=INDXT
GOTO 50
ENDIF
ENDIF
I=L
J=L+L
20 IF (J.LE.IR) THEN
IF (J.LT.IR) THEN
IF (YE(INDX(J)) LT YE(INDX(J+1))) J=J+1
ENDIF
IF (Q.LT.YE(INDX(J))) THEN
INDX(I)=INDX(J)
I=J
J=J+J
ELSE
J=IR+1
ENDIF
GOTO 20
ENDIF
INDX(I)=INDXT
GOTO 10
C
50 CONTINUE
C
DO 51 I=1,N
TEMP(I)=YE(I)
51 CONTINUE
DO 52 I=1,N
YE(I)=TEMP(INDX(I))
52 CONTINUE
C
DO 53 I=1,N
TEMP(I)=XE(I)
53 CONTINUE
DO 54 I=1,N
XE(I)=TEMP(INDX(I))
54 CONTINUE
C

```



```
DO 55 I=1,N
  TEMP(I)=PSVAL(I)
55 CONTINUE
  IF (JOB.EQ.1) THEN
    DO 56 I=1,N
      PSVAL(I)=TEMP(INDX(I))
56 CONTINUE
    ENDIF
    RETURN
  END
C
```

Appendix F

Experimental Observations on Normal Pressure and Shear Stress Distributions

Experimental observations on the normal pressure and shear stress distributions along the tire-terrain interface were reported by Oida et al [7].

Figure F.1 shows the variation of ratio of the maximum pressure to the normal load applied on the tire with slip, and that of the maximum shear stress. In Fig. F.2 the variation of the position of the maximum pressure with slip is presented.

Figure F.1: Variations of the Maximum Pressure/Weight and Shear Stress/Weight with Slip

Figure F.2: Variation of the Position of the Maximum Pressure with Slip

END

1 9-0 7-9 4

FIN

```

C
IF ((WIN1.EQ.1) OR (WIN2.EQ.1)) THEN
250 WRITE(' ') 'Anything to change ? (Y/N)'
  READ ('(A)') CHNG
  IF (CHNG.EQ.'Y') CHNG='Y'
  IF (CHNG.EQ.'N') CHNG='N'
  IF ((CHNG.NE.'Y') AND (CHNG.NE.'N')) GOTO 250
  IF ((JOB.EQ.2) AND (WIN1.EQ.1)) WRITE(' ',800)
  CALL GDRAWK(1)
  CALL GCLWK(1)
  CALL GCLKS
  ENDIF
C
IF ((JOB.EQ.1) THEN
  WRITE(' ') '[CRT Display]'
  IWIN2=0
  IWIN1=1
  ELSE
  WRITE(' ') '[PS File Creation]'
  IWIN1=0
  IWIN2=1
  ENDIF
C
  Open GKS Routines for General Case
C
  CALL OPNGKS(2)
  DO 1 I=1,4
    ASFLST(I)=GINDV
  1 CONTINUE
  CALL SETASF(ASFLST)
  CALL OPNPSW(0,IDEVS(JOB))
  CALL GAGIWP(1)
  CALL GSWF(1,XMIN,VXMAX,VYMIN,VYMAX)
  CALL GSWN(1,WXMIN,WXMAX,WYMIN,WYMAX)
  CALL GSELNT(1)
  CALL GSPLR(1,PLI,TYPE,WIDTH,COL)
  CALL GSTXFP(IFONT,1)
  CALL GSLWSC(1,0)
C
IF (CHNG.EQ.'N') GOTO 350
C
  select one of the following options for viewing
C
  Eye point : location of eye in the same coordinate
              system where the object is displayed
  Rotation angle: absolute angle with horizontal, rotate
                  the picture not the object
  Scale factor : scale the picture not the object
  Hidden       : ask whether hide those lines that are
                  not supposed to be seen.
                  The status will remain until next
                  selection is made. The default is No
C
WRITE (' ') '> Eye Point: x,y,z'
READ (' ') X1,Y1,Z1
IF ((CHNG.EQ.'Y') AND (JOB.EQ.2)) THEN
  WRITE (' ') '> Rotation angle'
  READ (' ') THETA
  STHETA=SIN(THETA*D2R)
  CTHETA=COS(THETA*D2R)
  WRITE (' ') '> Scale factor'
  READ (' ') FACTR
  WRITE (' ') '> Hidden ? (Y/N)'
  READ ('(A)') IH
  IF (IH.EQ.'Y') IH='Y'
  IF (IH.NE.'Y') IH='N'
  ELSE
    STHETA=0.0
    CTHETA=1.0
    FACTR=1.0
    IH='N'
  ENDIF
C
  Determine the visible surfaces and their
  normal vectors.
C
  300 CONTINUE
  CALL NORMAL
C
  Hidden Line Processing and Clipping
C
  360 CONTINUE
  IF (IH.EQ.'Y') THEN
    DO 37 I=1,NP
      IVIS(I)=0
    CALL HIDDEN
  ELSE
    IPL=IED
  ENDIF
C
  350 IF (JOB.EQ.2) THEN
    WRITE (' ') ' (PS File is prepared.)'
  ENDIF
C
  -- clear workstation
  CALL GCLRWK(1,1)
C
  DO 99 I=1,IPL
    --- assign lines
    IF (IH.EQ.'Y') THEN
      CMESH(1,1)=XE(I,1)
      CMESH(2,1)=YE(I,1)
      CMESH(1,2)=XE(I,2)
      CMESH(2,2)=YE(I,2)
    ELSE

```